CS410P

LAB 15L

Purpose

This assignment will be another tutorial on NumPy and a gentle introduction to matplotlib

Problem

You can create these sequence of instructions as part of one program, by printing out the result of each step (or wherever appropriate) to the output. Note that your output may differ from mine wherever random numbers are used.

No lists, no loops, no if statements allowed, since the whole point of using NumPy is to create arrays and make use of the functions that come with the package.

For every one of these items set a Python variable equal to the result and print out the variable value.

Import the numpy package under the name np

1. Generate two random matrices A and B of size 5 x 5 (random numbers from 1 to 10 only). Run a check to show if they are equal (every element of A with every element of B).

2. Create a random vector of size 50 and find the mean value

3. Create a random vector of size 100 and replace the maximum value by 0

4. Create two random vectors x and y each of size 50. Perform the following operations each time set a variable to contain the results each of these expressions and print the results.

$$
\begin{aligned}
s_x &= \sum_{i=0}^{n-1} x_i \\
s_y &= \sum_{i=0}^{n-1} y_i \\
s_{xy} &= \sum_{i=0}^{n-1} x_i y_i \\
s_{xx} &= \sum_{i=0}^{n-1} x_i^2 \\
m &= (s_x s_y - s_{xy} n)/(s_x^2 - s_{xx} n)) \\
b &= (s_y - s_x m)/n
\end{aligned}
$$

5. If you haven't guessed by now, we actually have everything we need to for solving the linear regression equation (using the method of least squares) to predict unknown quantities when we have a set of known pairs of values.

   $$f(x) = mx + b$$

   Compute an array using the above equation for all values of x much like you did for assignment 5P

6. Now compute and print the sum of the square of the residuals "**r**" using:

   $$\sum_{i=0}^{n-1} (y_i - f(x_i))^2$$

   Compute and print the total sum of squares "**t**" using:

   $$\sum_{i=0}^{n-1} (y_i - \bar{y})^2$$

   Compute and print the coefficient of determination using 1 – **(r/t)**

7. Use matplotlib to plot two simple plots (on the same figure window):

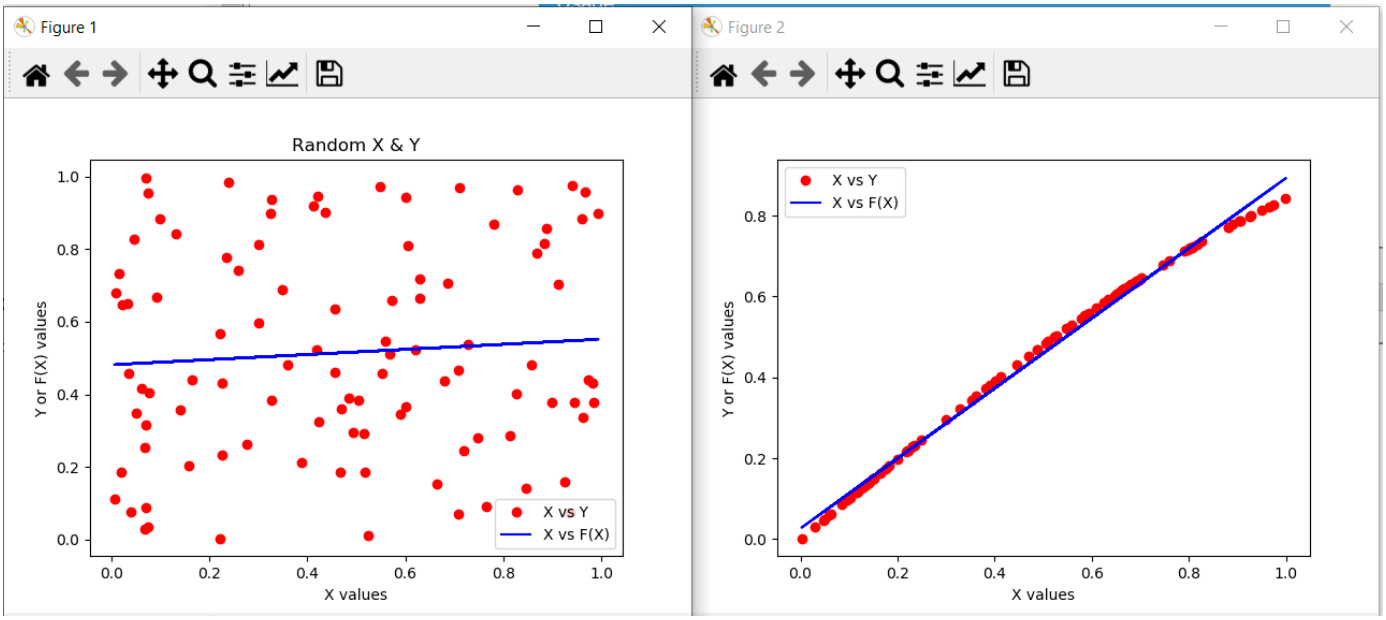
   1) Plot x vs y (x and y arrays/vectors as defined above in step 4 (use red dots)
   2) Plot x vs fx (x is the same array, fx is the computed (new y) array in step 5 (use blue dashes)

8. Repeat steps 4 through 7, this time create a vector x of 100 random values, and vector y which is a sine function of x i.e. we want to create a vector y which is related to x closely in some manner. When you plot the two graphs, use a new figure window so that the previous plots stay and do not get obscured.

For your plots you need to add axes labels, legend, title to make them more meaningful.

Output

Sample output is available in the public folder. Sample screen shots from the plots is shown below (your plots will look different than mine since we are using random numbers!



Grade Key

A	Items 1, 2, 3, 4, 5, 6, 8, 9, 10 (8 points each) (If loops are used: -50)	**72**
B	Items 7, 11 (Plotting) – 14 points each	**28**