CS410P

ASSIGNMENT 5P

Purpose

The purpose of this assignment is to have you work on a mathematical problem/analysis with lists and functions and file input, along with working with multiple files for creating your program/source code.

Scenario

When the relationship between two related quantities appears linear, a linear regression equation will give the best fit to that data. The context of this assignment is that you will compute a linear regression equation to fit a straight line to load deflection data for a mechanical coil spring. The computed equation will satisfy the least squares criterion. That is, it will minimize the sum of the squares of the deviations of the observed deflections from those predicted by the equation.

We start with a set of experimental data which plots the relationship between two quantities, the weight of a load on a spring and the corresponding compression of the spring, Fitting a curve to known data enables estimation of spring deflections for other loads not in the data. This curve can be the basis for calibrating a spring scale. The data consists of a collection of experimentally obtained pairs (x_i, y_i) with $i = 0, 1, 2, \ldots., n\text{-}1$. The x_i *s* (independent variables) are the weights; the y_i *s* (dependent variables) are the corresponding deflections. We want to find a (linear) function *f* such that $f(x_i) \approx f(y_i)$.

A) Least Squares Method

The method of least squares, a technique used for determining the equation of a straight line for a set of data pairs, will calculate coefficients *m* and *b* in the following linear regression equation:

$f(x) = mx + b$

When the original x_i values are substitued into the equation, estimate $f(x_i)$ values are obtained. For each x_i, the *residual* is defined to be the difference beween the observed y_i and the estimated $f(x_i)$. Thus in the least squares method the sum of the square of the residuals

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

will be minimized.

Algorithm

For the *n* data pairs (x_i,y_i) with $i = 0,, 1, \ldots., n\text{-}1$ the slope *m* and the intercept *b*, for the least squares linear approximation $f(x) = mx + b$ are calculated in the following way:

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

After printing the information about the data file and number of data points, your program will compute the *m* and *b* values, which are then use to compute the estimated $f(x_i)$ values and residuals.

Next we want to compute the sum of squared residuals, the total sum of squares and the coefficient of determination as determined by the following formula:

Sum of squared residuals (r):

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

Total sum of squares (t):

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

Coefficient of determination: 1-(r/t)

Once all the computations are done we want to print the entire set of computations for least squares approximation in a tabular format as shown in the sample output below.

B) Pearson Method

The Pearson Coefficient will be a number between -1 and 1, where a value close to 1 indicates good correlation, values close to 0 will indicate no correlation and values close to -1 indicates negative correlation. The Pearson method is not used for line fitting but just to indicate what kind of correlation exists between the pairs of (x,y) values.

Algorithm

For the *n* data pairs (x_i,y_i) with $i = 0,, 1, ..., n-1$ the Pearson Correlation Coefficient r is found by using the following formula:

The Pearson Coefficient will be a number between -1 and 1, where a value close to 1 indicates good correlation, values close to 0 will indicate no correlation and values close to -1 indicates negative correlation.

You can use either one of the two Pearson Coefficient formulas from the lecture notes (just showing you one below).

$$r = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n}}{\sqrt{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}} \sqrt{\sum_{i=1}^{n} y_i^2 - \frac{(\sum_{i=1}^{n} y_i)^2}{n}}}$$

For the Pearson method all we need to is to print the above "r" value which is the Pearson correlation coefficient which will be a number between -1 and 1.

Input

The input is assumed to be coming from a file. The input consists of a number of pairs of values, where each pair contains two floating point values which represent a load on the spring and the resulting compression distance, respectively. You should read the data into two lists one for x and one for y. These are "parallel" lists meaning that you have exactly the same number of items in both lists, and the corresponding element of each lists make up an (x, y) pair.

Output

Output from the Least Squares method is printed first followed by the Pearson method. While your output does not have to look exactly like mine but it should be as complete and as easy to read. For the Pearson method we are simply interested in seeing the final Pearson Correlation Coefficient.

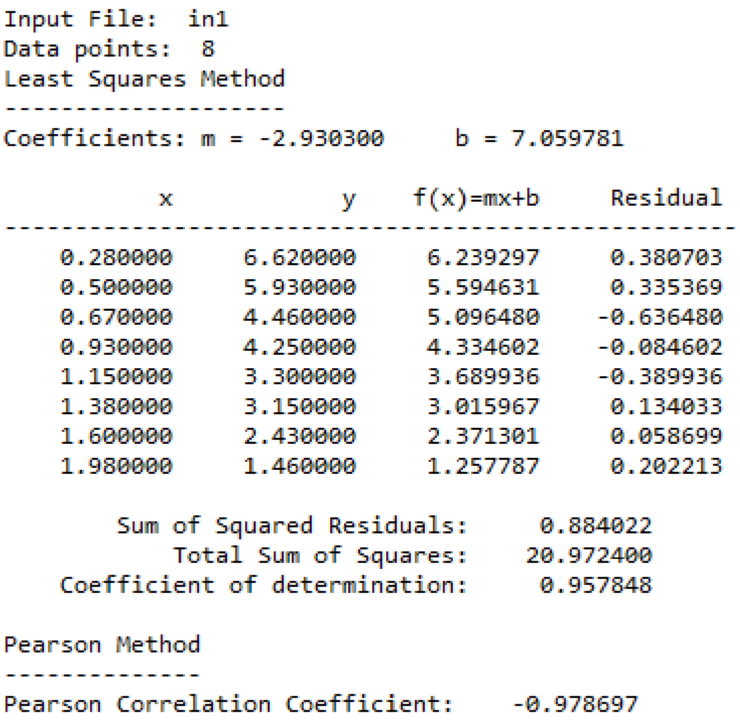
Requirements

- You are given the file containing the main() function (this is the file called regressions.py in the public folder). You need to write all the other functions in a file called regression_functions.py, you will need to submit only this file (do not change anything in regressions.py, just use it for your testing). Read the sequence of steps performed in the main() function before you proceed to implement other functions.
- Using the two lists containing the input (x, y) pairs, your program must eventually create two other lists – one which stores the corresponding f(x) value for each (x, y) pair, and another which stores the corresponding residual values for each (x, y) pair. In other words you end up having 4 "parallel lists" in your program.
- You cannot use any library (or numpy) functions for computing the regression coefficients for any methods. You will need to compute the equations in your program.
- You must use at least the following functions:
    - A function that reads in the data to the list (**argument**: data file name, **return values** are the two lists storing x and y values)
    - A function that computes both the y-intercept and slope (**arguments**: lists containing x and y values; **return values**: y-intercept and slope)
    - A function that computes two lists: f(x) and residuals (**arguments**: lists containing x and y values, slope and y-intercept (computed from the function above); **return values**: list containing f(x), list containing residuals
    - Create separate functions for computing the sum of squared residuals and total sum of squares and return the results appropriately, followed by a function that computes and returns the coefficient of determination
    - A function that computes the Pearson correlation coefficient (**arguments**: lists containing x and y values; **return value**: Pearson coefficient r)

  - Two functions to print results, one for Least squares approximation, one for Pearson.

- All function definitions must have at least have a block comment describing what they do.
- Other than printing the input file name and number of data points in the main function, all other printing must be done in two separate functions, one for printing Least squares results, and one for printing Pearson results.
- Other than your main function, all your other functions must be written in a separate Python file. The main() function seeks input from the user (for filename), calling the function to read data from the input file, followed by calling appropriate functions to compute and print the results.
- Work on the program using the modularity of the functions
- As always test your program carefully and follow good programming style.

Sample Output



```
Input File:  in1
Data points:  8
Least Squares Method
--------------------
Coefficients: m = -2.930300     b = 7.059781

           x            y     f(x)=mx+b     Residual
----------------------------------------------------
    0.280000     6.620000     6.239297     0.380703
    0.500000     5.930000     5.594631     0.335369
    0.670000     4.460000     5.096480    -0.636480
    0.930000     4.250000     4.334602    -0.084602
    1.150000     3.300000     3.689936    -0.389936
    1.380000     3.150000     3.015967     0.134033
    1.600000     2.430000     2.371301     0.058699
    1.980000     1.460000     1.257787     0.202213

        Sum of Squared Residuals:     0.884022
            Total Sum of Squares:    20.972400
    Coefficient of determination:     0.957848

Pearson Method
--------------
Pearson Correlation Coefficient:    -0.978697
```

Grade Key:

A	Input file read correctly with x and y lists updated	14
B	Initial printing of input file name and data set size done in main() but the rest of the printing done only in the appropriate functions	4
C	Accuracy of Least square results: slope, intercept, fx, residual, sum of squared residuals; total sum of squares, coefficient of determination	42
D	Accuracy of Pearson coefficient	20
E	Output for least squares approximation	16
F	Output for Pearson coefficient	4
G	Library functions must not be used for computing Least squares, Pearson coefficients. (Can be used for common Math or other functions like sqrt, fabs, abs, mean, etc). Cannot use Numpy arrays.	-35