Programming Project 08

This assignment is worth 50 points (4.0% of the course grade) and must be **completed and turned in before 11:59 on Monday, September 15, 2005.**

Assignment Overview
This assignment will give you more experience on the use of functions, lists and file manipulation. You will practice them by processing a file from a real-life dataset. In general, any time you find yourself copying and pasting your code, you should probably place the copied code into a separate function and then call that function.

Problem Statement
Given a data file of 420 cars from the model year 2004 (http://www.amstat.org/publications/jse/datasets/) , containing various pieces of information, create a linear regression model of the relationship between engine size and average miles per gallon.

Background

Linear regression is a form of regression analysis in which the relationship between one or more independent variables and another variable, called the dependent variable, is modeled by a least squares function, called a linear regression equation. A linear regression equation with one independent variable represents a straight line when the predicted value (i.e. the dependant variable from the regression equation) is plotted against the independent variable: this is called a simple linear regression. For example, suppose that a straight line is to be fit to the points (y_i, x_i), where $i = 1, \ldots, n$; y is called the **dependent variable** and x is called the **independent variable**, and we want to predict y from x.

Least Squares and Correlation
The method we are going to use is called the least squares method. It takes a list of x values and y values (the same number of each) and calculates the slope and intercept of a line that best matches those values. See http://easycalculation.com/statistics/learn-regression.php for an example.

To calculate the least squares line, we need to calculate the following values from the data:

- `sumX` and `sumY`: the sum of all the X values and the sum of all the Y values
- `sumXY`: the sum of the product of each corresponding X,Y pair
- `sumXSquared` and `sumYSquared`: the sum of the square of every X value and the square of every Y value
- `N`: the number of pairs

The calculation then is:

- `slope=(N*sumXY - (sumX*sumY))/(N*sumXSquared - (sumX)**2)`
- `intercept = (sumY – (slope*sumX)) / N`

We will also then calculate the correlation coefficient, and indication of how "linear" the points are (how much, in total, the points are correlated as a line). That calculation is:

- `corr = (N*sumXY - (sumX*sumY)) /`
  `sqrt((N*sumXSq - (sumX)**2) * (N*sumYSq - (sumY)**2))`

The correlation value ranges between -1 and 1. A negative value means an inverse correlation, a positive value a positive correlation. Values near -1 or 1 are "good" correlations, values near 0 are "bad" correlations. See http://easycalculation.com/statistics/learn-correlation.php

Project Description

- gather the data from the provided file `'04cars.data'`. The file `'04cards.txt'` describes the data. Engine size will be the `x` values, average mpg the `y` values. This must be done with a function.
    - Remember we want the average mpg (the average between highway and city mileage).
    - some data does not contain the required fields. Any missing data means that car must be skipped for the calculation.
- calculate the slope and intercept of a linear regression line through the data. Print those two values. This must be done with a function.
- calculate the correlation between the x and y data. Print the correlation. This must be done with a function.
- Plot the individual car entries using `matplotlib.`
- Plot the calculated regression line through the data.

Deliverables

proj08.py – your source code solution (remember to include your section, the date, project number and comments in your program).

1. Please be sure to use the specified file name, i.e. "proj08.py"
2. Save a copy of your file in your CS account disk space (H drive on CS computers).
3. Electronically submit a copy of the file.

Notes and Hints:

- Don't try to tackle this project all at once. Complete one function (or part of a function) and test it out.
- Test you least squares function on known data to make sure it works
- Matplotlib details. Look at the book chapter on angel, but remember:
    - import pylab
    - pylab.plot(xList, yList, options)
    - for options, you can select the color and the type of 'pip' that shows up in the plot such as 'ro' (red circles).
    - pylab.show() will show the plot. Be sure to plot everything (car values and lines) before you call show.
- You should ***test your functions*** before using them in the program. Create some small lists of known x and y values, for example [1,2,3,4,5] for both x and y. The slope and intercept of that should be obvious, as should the correlation. If you don't get the required answers, fix the function before moving on. Create a small cars file with only two or three entries and test that you can parse it correctly. Testing functions will make your life easier.