LAB 1 - Simple Linear Regression

In this lab, we will perform simple linear regression in Python. To do this, we will need the .csv provided in the assignment.

The .csv file contains information related to an imaginary soccer team (it is a bigger version of the .csv file we examined in the Python tutorial). We will investigate the "experience" and "salary" columns and see if there is a correlation between players' experience and their salary. Simple linear regression is going to be the method of choice for this task. Please read the following instructions:

- (10 pts) Read the .csv file, extract the "Experience" and "Salary" columns into two different arrays. The experience vector is going to be the x, and the salary vector is going to be our y, in the upcoming instructions. When extracting these values to vectors, don't use standard lists. We will use `numpy` arrays since it is much easier to do element-wise operations with them, unlike standard lists.

- (10 pts) Implement two different functions (which will be explained later). In the script (or main function), read the .csv file, extract the columns and call the corresponding functions in order. The script and the functions should be on the same file.

- (50 pts) The first function, named `simlin_coef`, takes in x and y as parameters, then computes and returns the regression coefficients. The eventual aim is to find two coefficients (b_0, b_1) such that:

  $$\hat{y} = b_1 x + b_0$$

  where the line $\hat{y}$ refers to our resulting model, and x is the experience vector. This line is later going to be plotted.

  The calculation of b_0 and b_1 consists of simple mathematical equations, where:

  $$b_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad b_0 = \bar{y} - b_1\bar{x}$$

  $\bar{x}$ and $\bar{y}$ refer to the *average* values of the x and y datasets, respectively.

- (30 pts) The second function named `simlin_plot` will plot the results. This function should take x, y, b_0 and b_1 as parameters and should do the following:

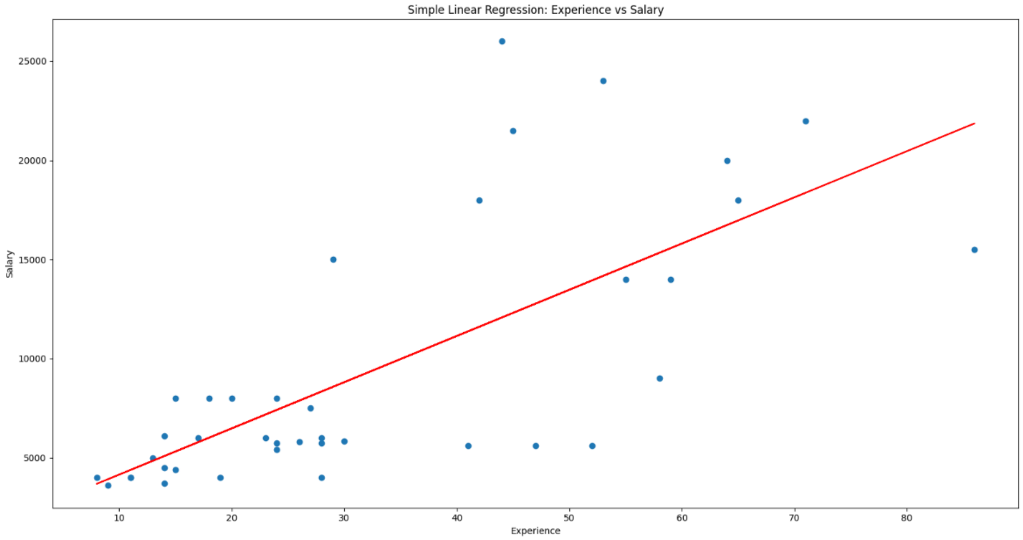
    - First, do a “scatter” plot using the actual data points: Use x as your x-axis and y as your y-axis for the graph. Use the blue color for this plot.

    - Next, draw the regression line on the same window: Use x as your x-axis and the $\hat{y}$ (the regression line, you have to calculate it first, using the first equation given in the previous page) as your y-axis. Use the red color for this *line* (not a scatter plot).

    - Last, call the `show` function to display your window.

IMPORTANT NOTE: The instructions specifically ask you to manually code the coefficient calculations. Bypassing these calculations (via using a library, `sklearn` being the most popular one) will mean that the calculations are not done, therefore that section (worth 50 points) will yield 0 points in total.

The result of your code should look like this:



Now, some insight regarding this lab session:

In this lab, we took our data, and produced the best possible linear model for it. This alone is not very useful though! In standard machine learning tasks, we need some data to *predict*.

For example, imagine that there is a new batch of data, but that new batch has its “experience” column empty. Now, looking at the “age” column of this new batch, we can predict what the “experience” values would be. If our linear model is accurate enough, the “experience” values should be very close to that “line” we drew earlier.

In the next lab, we will see more about this distinction.