LAB2 - Simple Linear Regression (continued)

In this lab, we will repeat our efforts to implement single linear regression on player age to experience (which was already done in LAB1), with small extra steps. This time, we will need two .csv files instead of one. First, we will have to *train* the algorithm using one of the datasets, i.e. we will use our simple linear regression algorithm on that dataset. Next, we will see how our estimation fares against the second dataset, we will *test* our regression line using that one. Two datasets to be used are given in the LAB2 folder.

Instructions:

- (10 pts) Extract the `"age"` and `"experience"` columns from the two .csv files just like you did in the first lab session. Label these columns as `"age_list_1"` and `"exp_list_1"` for one file, `"age_list_2"` and `"exp_list_2"` for the other.

- (20 pts) Perform the linear regression algorithm twice, using the data you extracted, ending up with coefficients. Label these coefficients accordingly (if you used `"age_list_1"` and `"exp_list_1"`, then label your *slope* as `"m1"`, and your *intercept* as `"b1"`. If you used the other set, label as `"m2"` and `"b2"`).

- (35 pts) Plot the regression lines:

  - In one window, make a scatter plot using `"age_list_1"` as your x-axis and `"exp_list_1"` as your y-axis. On the same window, draw the regression line. When calculating this line (i.e. $y = mx + b$), use `"age_list_1"` as your "x" **but use `"m2"` and `"b2"` as your coefficients!** Label this line as `"exp_pred_1"`. Plot `"age_list_1"` as your x-axis, and use `"exp_pred_1"` as your y-axis.

  - *In a separate window*, do the same thing with the other dataset. Again, make a scatter plot using `"age_list_2"` as your x-axis and `"exp_list_2"` as your y-axis. Calculate and draw the regression line on the same window, using `"age_list_2"` as your "x" **and with `"m1"` and `"b1"` as your coefficients.**

    - When plotting, make sure that both plots appear on the screen *at the same time*! Make sure that you call the *`show()`* function only once in the end.

- (35 pts) As a final extra step, implement a method which **calculates** and **displays** the R^2 score. This method should take two parameters: One for the actual y values, the other for the estimated y values. The calculation can be found here:

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

  where

$$\mathrm{RSS} \quad = \quad \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad \mathrm{TSS} = \sum (y_i - \bar{y})^2$$

  Here:

- y_i refers to the ith actual experience value,

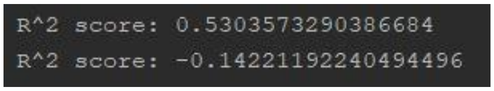
- $\hat{y}_i$ refers to the ith prediction value

- $\bar{y}$ refers to the *average* of all actual experience values.

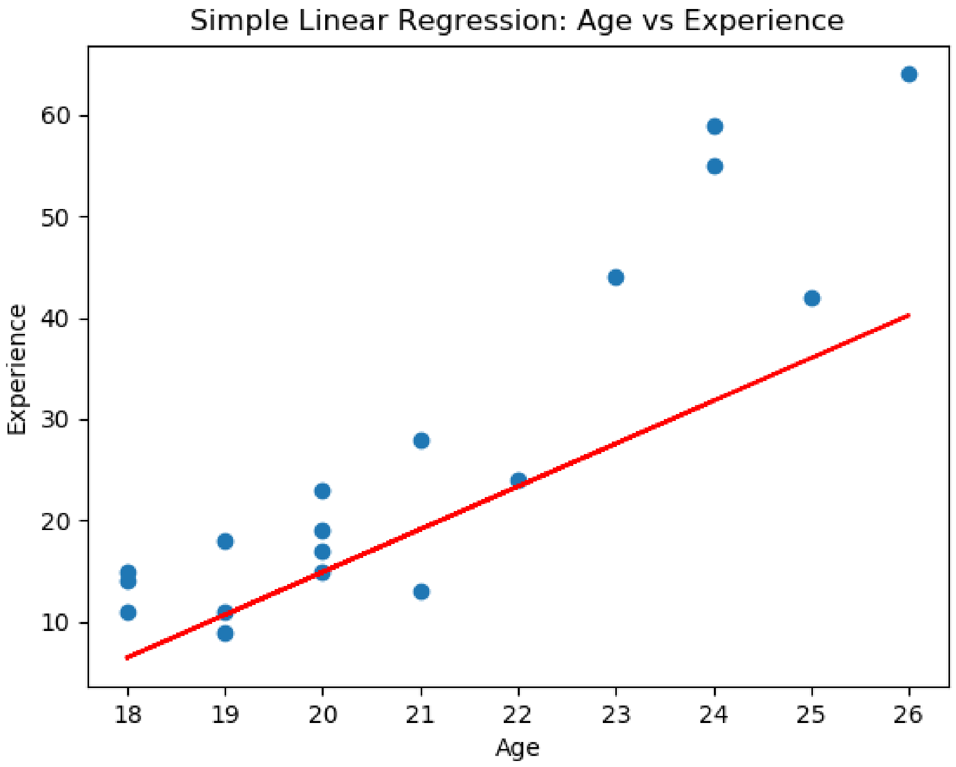
  You will call this method twice.
  One call will be done by using `"exp_list_1"` and `"exp_pred_1"` as parameters.
  The other call will be done by using `"exp_list_2"` and `"exp_pred_2"` as parameters.

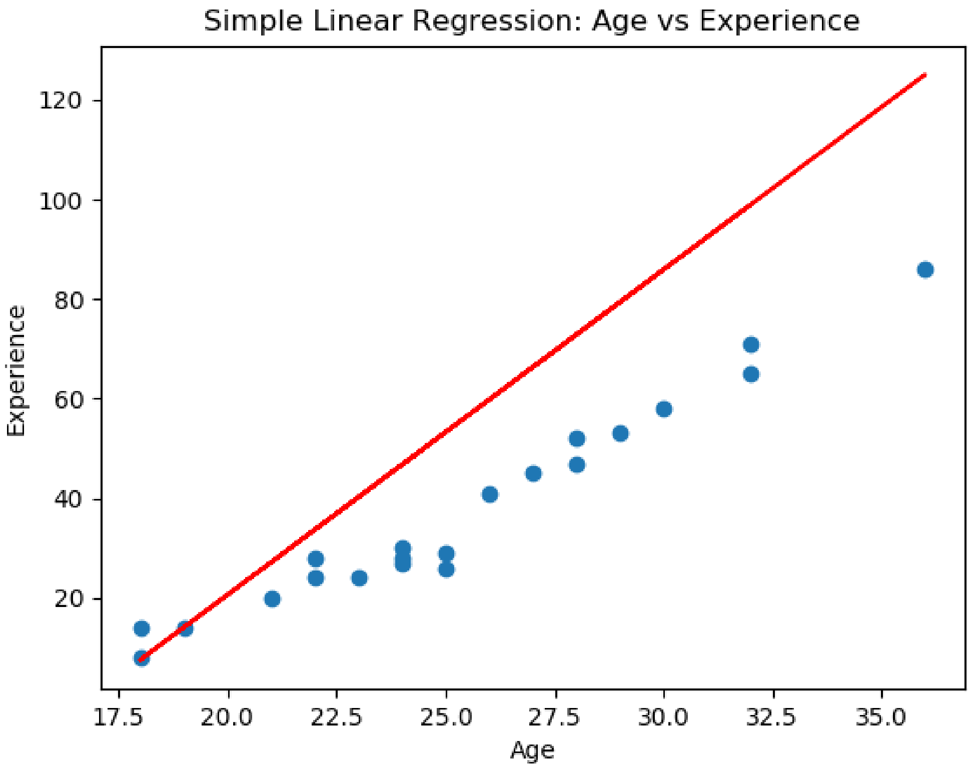
- **WARNING: The only outside packages available for us to use will be "pandas", "numpy" and "matplotlib". Any other packages are not permitted for use.**

Below are output samples for the plots and for R^2 scores:



```
R^2 score: 0.5303573290386684
R^2 score: -0.14221192240494496
```


Simple Linear Regression: Age vs Experience
Experience
60
50
40
30
20
10
18
19
20
21
22
23
24
25
26
Age


Simple Linear Regression: Age vs Experience
Experience
120
100
80
60
40
20
17.5
20.0
22.5
25.0
27.5
30.0
32.5
35.0
Age

Continuing with the insight on this lab session:

As mentioned, in the previous lab, we built a model, but we never tested that model. In standard machine learning tasks, a model is usually tested on a different data, to see how accurate our model is. In this lab, we can see the distinction.

When we create a model using a data, we *train* the data. The data we use is called the *train* set. Then, we compare our model to a different dataset, which is called the *test* set.

Usually, a dataset is divided into train and test sets, in order to create a more accurate model overall. We have 2 different .csv files here, which act as train and test sets, so we don’t have to divide our data, it’s already divided. In future labs, we will do the dividing ourselves.

Showing the results in the form of a numerical value is also very important. Here, we were able to show the results by directly plotting the regression line and the data itself, but in many cases, the data (or the model) will be in a multiple-dimension representation, so plotting the data will be impossible. In these cases, we need numerical values to see if the model is accurate or not.

We have calculated the R^2 scores for two different cases. This metric compares the estimation (the regression line) with the mean of the data (which is not shown on the graphs, but you can imagine it as a horizontal line at the average value of the y-axis). The score reflects how much of the variance we were able to reduce using our model.

In the second case, the R^2 score is negative. This means that the mean of the data has lower variance than our model, which means that our model increased the variance.