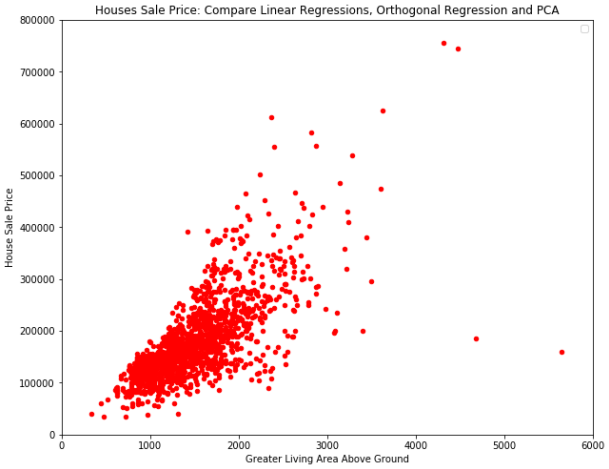
Exercise 2 - April 26th
Principal Components or Regression?

The goal of this exercise is, using a two-dimensional dataset, to compare Principal Component Analysis (PCA) – an unsupervised learning approach - with Linear Regression – a supervised learning approach.
The dataset used can be obtained from a Kaggle competition named "House Prices: Advanced Regression Techniques". It contains 1460 training data points and 80 features that might help predict the selling price of a house. The dataset is described in details in https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data. A copy of the dataset, called *houseprices.csv*, is also available in the course folder.

For this exercise we will focus on the two following variables of the file *houseprices.csv*:
Saleprice: the house sale price ($)
GrLivArea: above ground living area (square feet)
We will compare the result of applying regression or PCA to this two-variable dataset. We will use the variable name x for *GrLivArea* and y for *Saleprice*. The following is the cross-plot of y vs x associated with the 1460 houses in the dataset.
A Python solution code is associated with the exercise, it is called *PCARegression.py*.

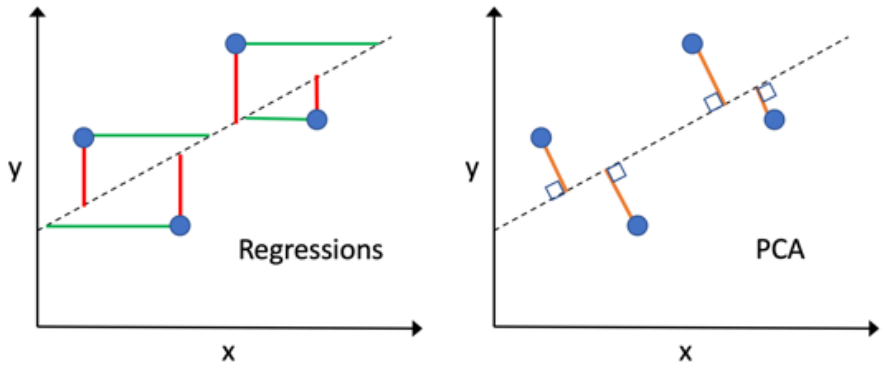


1. Give the mathematical expression of the bias term θ_0 and the slope θ_1 of the regression line for predicting y from x.
   Calculate the regression parameters θ_0 and θ_1.
   Also calculate the regression parameters and the R2 variance score using the *LinearRegression* module in the *sklearn.linear_model* library . Check that your calculations of θ_0 and θ_1 are right.

2. Give the mathematical expression of the bias term θ_0' and the slope θ_1' of the regression line for predicting x from y.
   Calculate θ_0' and θ_1' .
   Also calculate the regression parameters and the R2 variance score using the *LinearRegression* module in the *sklearn.linear_model* library . Check that your calculations of θ_0' and θ_1' are right.

3. Plot the two regression lines calculated in the previous questions, and cross-plot the dataset on the same figure.

4. With regression, as seen in the two previous questions, what is minimized is the sum of squares of the differences between the data values and the regression lines. This can be summarized in the picture below (the sum of the squares of the red lines are minimized by regression of y vs x, the sum of the squares of the green lines are minimized by regression of x vs y). On the other hand, PCA minimizes the sum of the squares of the orange lines, that is the perpendiculars to the calculated principal axis (we can also say, as discussed at the end of the morning, that PCA maximizes the variance of the data projected on the principal axis).



There are two ways to calculate the slope of the PCA line in two dimensions (it is enough to calculate the slope as the second parameter can be derived from the fact that, as was the case for the two regression lines, the PCA line goes through the point associated with the mean of the two variables).
The first approach is to calculate it analytically using the method known as Orthogonal Regression, where the sum of the squares of the orange segments in the above figure is minimized. It can be demonstrated (see: Orthogonal Regression: a Teaching Perspective" by James R. Carr) that the formula for the slope is:

$$\theta_1'' = \frac{(\sum_{i=1}^{m} V_i^2 - \sum_{i=1}^{m} U_i^2) + \sqrt{(\sum_{i=1}^{m} V_i^2 - \sum_{i=1}^{m} U_i^2)^2 + 4(\sum_{i=1}^{m}(U_i V_i)^2)}}{2\sum_{i=1}^{m}(U_i V_i)}$$

Where U and V are the centered variables x and y, that is the variables x and y after substraction of their mean:

$$U = x - m_x \text{ and } V = y - m_y$$

The second approach is to calculate it using (for instance) the *PCA* module in the *sklearn.decomposition* library. With this second approach, it is important to first standardize *Saleprice* and *GrLivArea*. Use the two approaches and verify that you obtain the same result.

5. Compare the three lines on the same cross-plot. What do you observe?