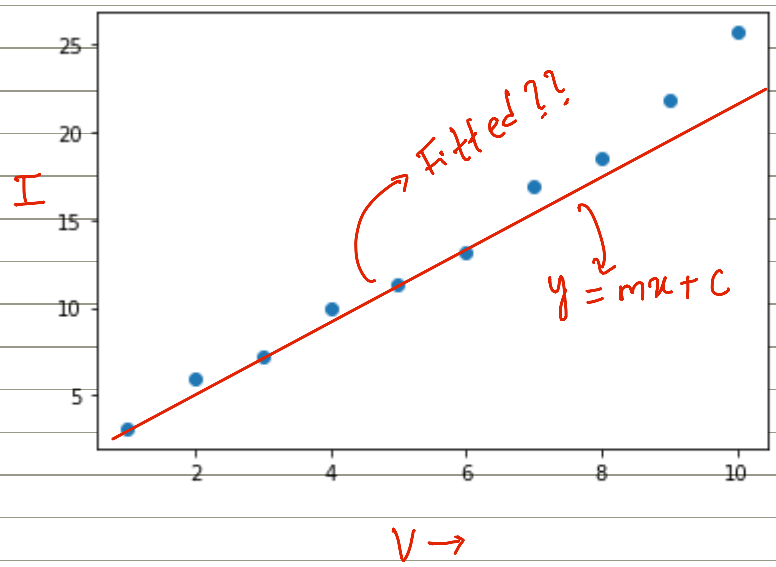
Curve Fitting

(Regression Analysis)

→ Suppose, we have a set of data points (x, y)

x (V)	(I) y
1	3.1
2	5.9
3	7.2
4	10.0
5	11.3
6	13.2
7	16.9
8	18.5
9	21.8
10	25.7

Scatter Plot



$V = IR$, $I/V = \frac{1}{R} = \text{slope}$

The curve of fitting of this data points, means, we need to find a function that best represents the data by minimizing the difference between the actual data & the predicted values from the function.

(1) Least Square Method :-

(i) Linear fit :-

In linear fit, we fit the data set with a straight line. The equation of straight line is $y = a_0 + a_1 x$. The given data points may or may not fall on the assumed line. In, general, the points may be scattered on the either side of the line.

The error function:

$$\varepsilon(a_0, a_1) = \sum_{i=1}^{n} (a_0 + a_1 x_i - y_i)^2 \quad \text{—(1)}$$

means

$$\left[\sum_{i=1}^{n} \left(y_i^{\text{predicted}} - y_i^{\text{actual}} \right)^2 \right]$$

For the best fit, this error should be minimum with respect to the chosen parameters.

Therefore, $\frac{\partial \varepsilon}{\partial a_0} = 0$ and $\frac{\partial \varepsilon}{\partial a_1} = 0$

Now,

$$\frac{\partial \varepsilon}{\partial a_0} = \frac{\partial}{\partial a_0} \left[\sum_{i=1}^{n} (a_0 + a_1 x_i - y_i)^2 \right] = 0$$

$$\Rightarrow \sum_{i=1}^{n} \frac{\partial}{\partial a_0} (a_0 + a_1 x_i - y_i)^2 = 0$$

$$\Rightarrow \sum_{i=1}^{n} 2(a_0 + a_1 x_i - y_i) = 0$$

$$\Rightarrow \boxed{n a_0 + a_1 \sum_i x_i = \sum_i y_i} \quad \text{—(2)}$$

Also, $\frac{\partial \varepsilon}{\partial a_1} = 0$, $\frac{\partial}{\partial a_1} \left[\sum_{i=1}^{n} (a_0 + a_1 x_i - y_i)^2 \right] = 0$

$$\Rightarrow \sum_i 2(a_0 + a_1 x_i - y_i) \cdot x_i = 0$$

$$\Rightarrow \boxed{a_0 \sum_i x_i + a_1 \sum_i x_i^2 = \sum_i x_i y_i} \quad \text{— (3)}$$

From eq^n (2) & (3),

$$\boxed{\begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}} \quad \text{— (4)}$$

$$\Rightarrow \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

$$= \frac{1}{n \sum_i x_i^2 - \left(\sum_i x_i\right)^2} \begin{bmatrix} \sum_i x_i^2 & -\sum_i x_i \\ -\sum_i x_i & n \end{bmatrix} \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{\sum_i x_i^2 \sum_i y_i - \sum_i x_i \sum_i x_i y_i}{n \sum_i x_i^2 - \left(\sum_i x_i\right)^2} \rightarrow a_0 \\[2em] \dfrac{-\sum_i x_i \sum_i y_i + n \sum_i x_i y_i}{n \sum_i x_i^2 - \left(\sum_i x_i\right)^2} \rightarrow a_1 \end{bmatrix} \quad \text{— (5)}$$

So, we can compute slope (a_1) & the intercept (a_0) from the above formula to get the expression for the fitting curve, $y = a_0 + a_1 x$.

(ii) <u>Non-linear fit</u> :-

For, the least square fit with polynomial of higher degrees, we can proceed in the same way. For ex:

for a polynomial of degree (2) : (quadratic fit)

$$y = a_0 + a_1 x + a_2 x^2 \quad —(6)$$

after going through similar calculations :

$$\varepsilon(a_0, a_1, a_2) = \sum_i (a_0 + a_1 x_i + a_2 x_i^2 - y_i)^2$$

$$\frac{\partial \varepsilon}{\partial a_0} = 0, \quad \frac{\partial \varepsilon}{\partial a_1} = 0, \quad \frac{\partial \varepsilon}{\partial a_2} = 0$$

We can obtain,

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} n & \Sigma x & \Sigma x^2 \\ \Sigma x & \Sigma x^2 & \Sigma x^3 \\ \Sigma x^2 & \Sigma x^3 & \Sigma x^4 \end{pmatrix}^{-1} \begin{pmatrix} \Sigma y \\ \Sigma xy \\ \Sigma x^2 y \end{pmatrix}$$

$\Downarrow$

Solve this eqn, to get a_0, a_1 & a_2.

Now, Let's see some python programs to implement these methods.