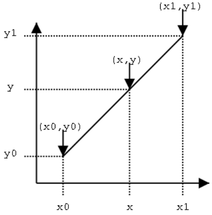
# Interpolation and data fitting

Commonly, in typical experiments, we know $f(x_i)$ at various $x_i$, not necessarily equally spaced. Additionally, we may not quite know the analytical expression of $f(x)$ that describes the data and allow to calculate its value at arbitrary $x \neq x_i$.

If the desired $x$ is/are between largest and smallest $x_i$'s, the problem is of *interpolation*. If outside, then it is *extrapolation*.

*Extrapolation* is a terribly risky endeavor unless backed up by solid theoretical idea(s). Here we study only *interpolation*.

*Interpolating* $(x, y)$ between $(x_0, y_0)$ and $(x_1, y_1)$ with a straight line,




$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \Rightarrow y = y_0 + \frac{x - x_0}{x_1 - x_0}(y_1 - y_0)$$

Generalize this idea by considering a polynomial of degree $N$ to approximate a function from a set of $N+1$ points

$$P_N(x_i) \equiv f(x_i) = y_i \quad \text{where } i = 0, 1, \ldots, N$$

The general form of $P_N(x)$ is

$$\begin{aligned}
P_N(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + \\
&\quad a_N(x - x_0)(x - x_2)\ldots(x - x_{N-1}) \\
P_0(x_0) &= a_0 = y_0 = f(x_0) \\
P_1(x_1) &= a_0 + a_1(x_1 - x_0) = y_1 = f(x_1) \\
P_2(x_2) &= a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = y_2 = f(x_2) \quad \text{etc.}
\end{aligned}$$

For linear interpolation $N = 1$,

$$a_0 = y_0, \quad a_1 = \frac{y_1 - y_0}{x_1 - x_0} \Rightarrow P_1(x) = y(x) = y_0 + \frac{x - x_0}{x_1 - x_0}(y_1 - y_0)$$

A particularly useful form, although looks complicated, is

$$P_1(x) = y_0 - \frac{x - x_0}{x_1 - x_0} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1$$

$P_1(x)$ is the polynomial of degree 1 for $1+1$ data points.

Generalized interpolation formula by Lagrange

$$P_N(x) = \sum_{i=0}^{N} \prod_{k\neq i} \frac{x - x_k}{x_i - x_k}\, y_i$$

As an example, 2nd order polynomial for 3 data points $x_0, x_1, x_2$

$$P_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\, y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}\, y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}\, y_2$$

**Ex 1.** Consider the following data table and show the $f(x=4) = 20$

| $x$ | 2 | 3 | 5 | 8 | 12 |
|---|---|---|---|---|---|
| $f(x)$ | 10 | 15 | 25 | 40 | 60 |

**Ex 2.** To understand what is going on, take a smaller data set

| $x$ | 0 | 10 | 20 | 30 |
|---|---|---|---|---|
| $f(x)$ | -250 | 0 | 50 | -100 |

Each term in Lagrange formula *i.e.* coefficient of $y_i$

$$i = 1 \qquad \frac{(x-10)(x-20)(x-30)}{(0-10)(0-20)(0-30)} = -\frac{x^3}{6000} + \frac{x^2}{100} - \frac{11x}{60} + 1$$

$$i = 2 \qquad \frac{(x-0)(x-20)(x-30)}{(10-0)(10-20)(10-30)} = \frac{x^3}{2000} - \frac{x^2}{40} + \frac{3x}{10}$$

$$i = 3 \qquad \frac{(x-0)(x-10)(x-30)}{(20-0)(20-10)(20-30)} = -\frac{x^3}{2000} + \frac{x^2}{50} - \frac{3x}{20}$$

$$i = 4 \qquad \frac{(x-0)(x-10)(x-20)}{(30-0)(30-10)(30-20)} = \frac{x^3}{6000} - \frac{x^2}{200} + \frac{x}{30}$$

Multiply each line with the corresponding $f(x_i)$ and add together the terms of like power to obtain

$$y(x) \equiv P_3(x) = -x^2 + 35x - 250$$

Lagrange's formula gives a simple quadratic description of the data.

If required, we can calculate, say, $f(x = 15) = 50$ either using the quadratic formula or directly from Lagrange's formula.

**DIY :** Find both way $f(x = 1969)$ from the data

| $x$ | 1951 | 1961 | 1971 |
|---|---|---|---|
| $f(x)$ | 2.8 | 3.2 | 4.5 |

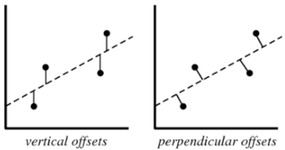
# Least square fitting

Set of data points generated $(x_i, y_i)$ are expected to be described by a **known** function $f(x)$ but with undetermined coefficients.

**Examples** – stress-strain data points within elastic limit, voltage-current data points. Task is to determine modulus of elasticity and Resistance of the material under consideration.

$$V = \mathrm{R}\, i \quad \text{and} \quad \text{stress} = \text{k} \times \text{strain}$$

Least square fitting is a way to do this and for that we need *sum of squares of the offset (residual)* of the data points from the curve $f(x)$. Offsets can be either *vertical* or *perpendicular*




Vertical offsets allow uncertainties of data points along $x$, $y$-axes to be independent of each other.

Why squares and not absolute values?

Squares permit offsets to be treated as continuous differentiable quantities.

Absolute values result in discontinuous derivatives.

Fit $N$ data points $(x_i, y_i, \sigma_i)$ to a model $f(x_i; a_1, a_2, \ldots, a_M)$ with $M$ parameters ($M < N$). Least square suggests to minimize squares of weighted offsets with respect to the parameters $a_k$'s

$$\chi^2 = \sum_{i=1}^{N} \left( \frac{y_i - f(x_i; a_1, a_2, \ldots, a_M)}{\sigma_i} \right)^2 \rightarrow \frac{\partial \chi^2}{\partial a_j} = 0$$

*Linear regression* : model the data points with a straight line

$$f(x) = a_1 + a_2 x \;\Rightarrow\; \chi^2 = \sum_{i=1}^{N} \left( \frac{y_i - a_1 - a_2 x_i}{\sigma_i} \right)^2$$

Minimizing $\chi^2$ w.r.t. $a_1, a_2$ yields

$$\frac{\partial \chi^2}{\partial a_1} = -2 \sum_{i=1}^{N} \frac{y_i - a_1 - a_2 x_i}{\sigma_i^2} = 0$$

$$\frac{\partial \chi^2}{\partial a_2} = -2 \sum_{i=1}^{N} \frac{x_i (y_i - a_1 - a_2 x_i)}{\sigma_i^2} = 0$$

Introducing a shorthand symbol $\sum_{i=1}^{N} \equiv \sum_i$, we ended up with

$$\sum_i \frac{y_i}{\sigma_i^2} = a_1 \sum_i \frac{1}{\sigma_i^2} + a_2 \sum_i \frac{x_i}{\sigma_i^2} \qquad \mathcal{S}_y = a_1 \mathcal{S} + a_2 \mathcal{S}_x$$

$$\equiv$$

$$\sum_i \frac{x_i y_i}{\sigma_i^2} = a_1 \sum_i \frac{x_i}{\sigma_i^2} + a_2 \sum_i \frac{x_i^2}{\sigma_i^2} \qquad \mathcal{S}_{xy} = a_1 \mathcal{S}_x + a_2 \mathcal{S}_{xx}$$

If the data errors $\sigma_i$ are not given, take $\sigma_i = 1$. However, your code must include $\sigma_i$ in case they are provided.

The solution of these equations are absolutely straight forward,

$$a_1 = \frac{\mathcal{S}_{xx}\mathcal{S}_y - \mathcal{S}_x\mathcal{S}_{xy}}{\Delta}, \quad a_2 = \frac{\mathcal{S}_{xy}\mathcal{S} - \mathcal{S}_x\mathcal{S}_y}{\Delta} \text{ where } \Delta = \mathcal{S}\mathcal{S}_{xx} - \mathcal{S}_x^2$$

Estimating the errors $\sigma_{a_i}$ on the parameters $a_i$ determined above

$$\sigma_{a_i}^2 = \sum_i \sigma_i^2 \left(\frac{\partial a_i}{\partial y_i}\right)^2$$

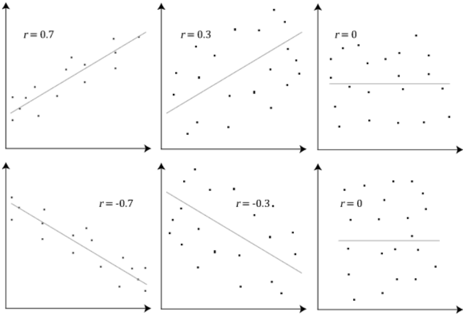
$$\Rightarrow \qquad \frac{\partial a_1}{\partial y_i} = \frac{\mathcal{S}_{xx} - \mathcal{S}_x\, x_i}{\Delta\, \sigma_i^2}, \quad \frac{\partial a_2}{\partial y_i} = \frac{\mathcal{S}\, x_i - \mathcal{S}_x}{\Delta\, \sigma_i^2}$$

$$\Rightarrow \qquad \sigma_{a_1}^2 = \frac{\mathcal{S}_{xx}}{\Delta} \quad \text{and} \quad \sigma_{a_2}^2 = \frac{\mathcal{S}}{\Delta}$$

To this end, in order to get some idea about how *good* is the **linear** fit we define an estimate called *Pearson's correlation coefficient* $r$

$$r^2 = \frac{\mathcal{S}_{xy}^2}{\mathcal{S}_{xx}\mathcal{S}_{yy}} \quad \text{where} \quad 0 \leq r^2 \leq 1$$

As $r^2 \to 1$ fit gets better.




**NB :** But is the straight line model itself good to fit the data? To address this requires ideas of *goodness of fit*, *confidence level*, testing some *null hypothesis* against $\chi^2$ probability distribution and so on.

The straight line model is generic enough for use in a few other models which can be reduced to straight line form usually by taking logarithms,

$$
\begin{aligned}
\text{exponential :} &\quad f(x) = a\, e^{b\,x} \;\rightarrow\; \log f(x) = \log a + bx \\
\text{logarithm :} &\quad f(x) = a + b \log x \\
\text{power law :} &\quad f(x) = a\, x^{b} \;\rightarrow\; \log f(x) = \log a + b \log x
\end{aligned}
$$

## Least square polynomial fitting

Polynomial model $f(x) = \sum_{k=0}^{n} a_k x^k$ can also be subjected to linear fitting. For simplified expressions, we put all the errors $\sigma_i^2 = 1$ without any loss of generality. In such case, $\chi^2 \rightarrow R^2$

$$
\begin{aligned}
R^2 &= \sum_{i=1}^{n} \left[ y_i - \left( a_0 + a_1 x_i + \cdots + a_k x_i^k \right) \right]^2 \\
\frac{\partial R^2}{\partial a_0} &= -2 \sum \left[ y_i - \left( a_0 + a_1 x_i + \cdots + a_k x_i^k \right) \right] = 0 \\
\frac{\partial R^2}{\partial a_1} &= -2 \sum \left[ y_i - \left( a_0 + a_1 x_i + \cdots + a_k x_i^k \right) \right] x_i = 0 \\
&\cdots \\
\frac{\partial R^2}{\partial a_k} &= -2 \sum \left[ y_i - \left( a_0 + a_1 x_i + \cdots + a_k x_i^k \right) \right] x_i^k = 0
\end{aligned}
$$

Set of equations for $a_i$'s are,

$$\begin{aligned}
a_0 n + a_1 \sum x_i + \cdots + a_k \sum x_i^k &= \sum y_i \\
a_0 \sum x_i + a_1 \sum x_i^2 + \cdots + a_k \sum x_i^{k+1} &= \sum x_i y_i \\
&\cdots \\
a_0 \sum x_i^k + a_1 \sum x_i^{k+1} + \cdots + a_k \sum x_i^{2k} &= \sum x_i^k y_i
\end{aligned}$$

which when written in matrix form becomes a problem of matrix inversion, and you are free to use your favorite inverter.

$$\begin{pmatrix} n & \sum x_i & \cdots & \sum x_i^k \\ \sum x_i & \sum x_i^2 & \cdots & \sum x_i^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_i^k & \sum x_i^{k+1} & \cdots & \sum x_i^{2k} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^k y_i \end{pmatrix}$$

$$\Rightarrow \quad \mathrm{X} \cdot \mathrm{a} = \mathrm{Y} \quad \rightarrow \quad \mathrm{a} = \mathrm{X}^{-1}\mathrm{Y}$$

# DIY fitting

1. A time versus angular velocity data set on deceleration of a rotating disc is in the file `fit1.dat`. Fit it with the following functions (*i*) $\omega(t) = \omega_0 + \omega_c t$ and (*ii*) $\omega(t) = \omega_0 \, e^{-\omega_c t}$. Determine $\omega_0, \omega_c$ and the quality of fit for both the functions by *Pearson's* $r$.
2. Distance ($r$) versus height ($h$) of the trajectory of a test missile is given in the datafile `fit2.dat`. Try quadratic fit $h = a_0 + a_1 r + a_2 r^2$ and determine the highest point reached by the missile.