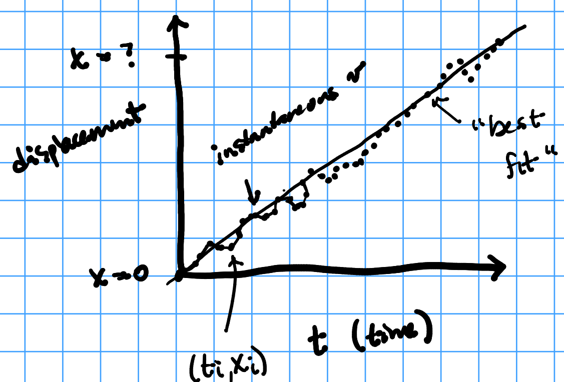
Lecture 17 - Least Square Fitting

I. Least Square Fits
II. " " " in Excel
III. " " " in Python

0. Numerical Calculus

- Fitting & Interpolation
- Integration
- simple differential equations (no need ODEs / PDEs)

I. Least Square Fitting



* Data from ~ jog

$$x = v\,t$$

↑ disp ↑ vel. ↑ time

* we only care about the average behavior & not the noise (assuming the model isn't wrong)

$$\check{x}_i = v\,\check{t}_i + \varepsilon_i$$

$\check{x}_i$ ← counter for the data pts $0, 1, \dots, n-1$

$v\,t_i$ ← ? — prediction for x

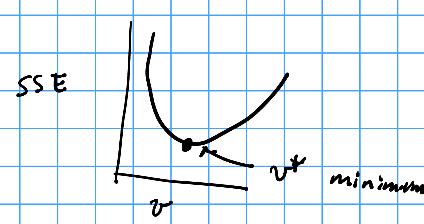
ε_i ← random noise — unimportant

* Least Square Fitting

- vary the free parameter (in this case is v) to minimize the total error from noise.

$$\varepsilon_i = x_i - v\,t_i$$

$$SSE = \sum_i (\varepsilon_i)^2 = \sum_i (x_i - v\,t_i)^2 = SSE(v)$$



- This is like solving a NLE w/ optimization

$SSE \neq 0$, just small

Too many variables to get $SSE = 0$, overdetermined system. (many Eqs not enough unknowns, only v)

General Formulation

$$\varepsilon_i = y_i - f(x_i)$$

model (prediction)

data

$$SSE = \sum_i \varepsilon_i^2 = \sum_i (y_i - f(x_i))^2$$

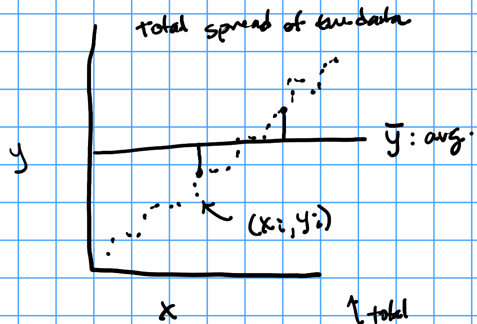
$$SSE(\xi) = \sum_i (y_i - f(x_i;\xi))^2$$

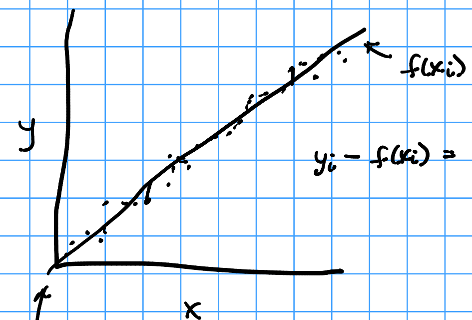
minimize SSE w.r.t. ξ.

* Goodness of Fit

- How well does our model "fit" or explain the data?

R^2: coefficient of determination





total spread of data

total error of model

$$SST = \sum_i (y_i - \bar{y})^2$$

$$SSE = \sum_i (f(x_i) - y_i)^2 = \sum_i \varepsilon_i^2$$

$$R^2 = \frac{SST - SSE}{SST}$$

SSE is big $R^2 \approx \frac{SST - SST}{SST} \approx 0$ Bad fit

SSE is small $R^2 \approx \frac{SST - 0}{SST} \approx 1$ Good fit

II. Fitting in Excel $\longrightarrow$ 2 ways

III. Fitting in Python

* polynomials : $f(x) = x^3 + x^2 - 3x + 2$ powers.

* Not polynomials