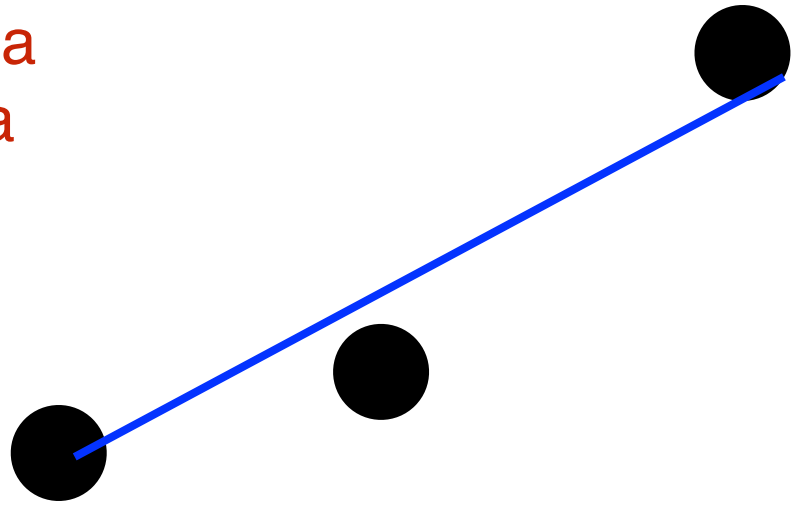
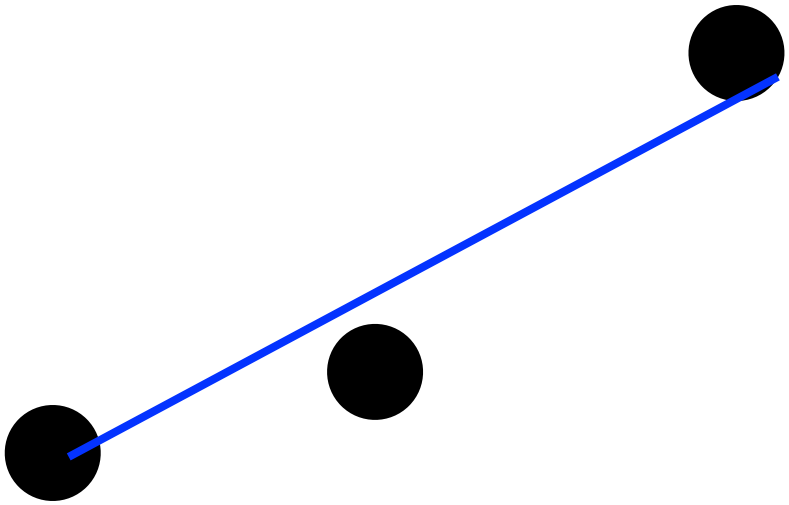
Some familiar slides

Imagine we have some data points, and we want to fit a straight line to them

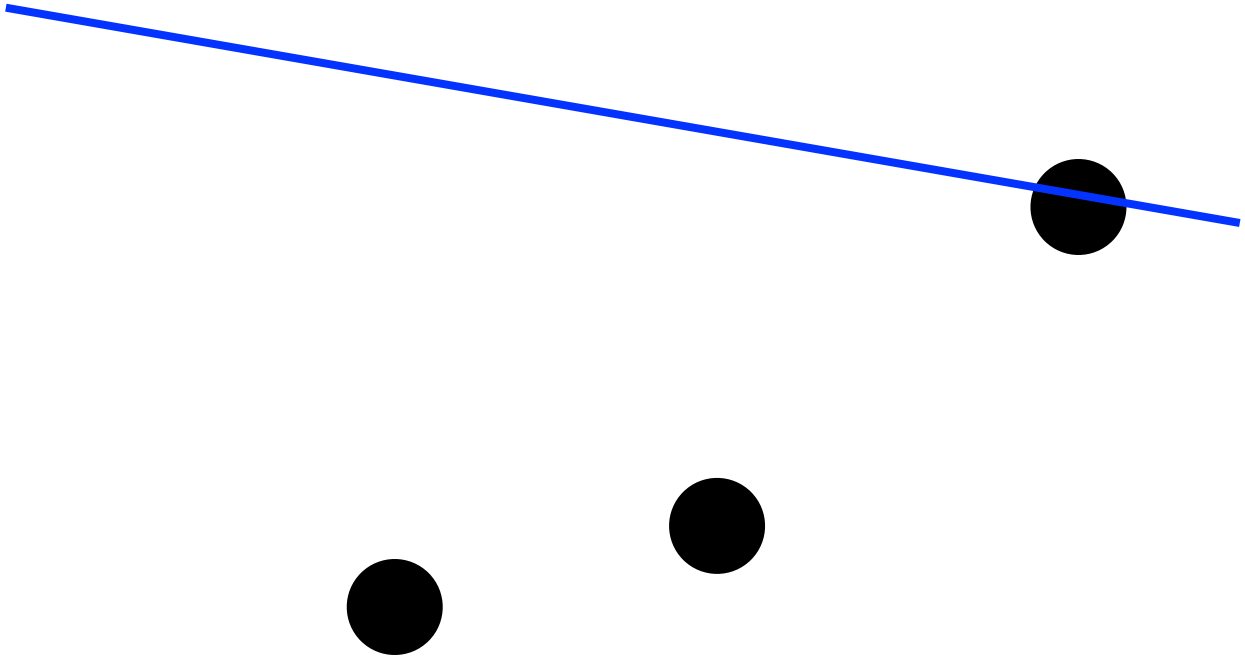


Linear fits



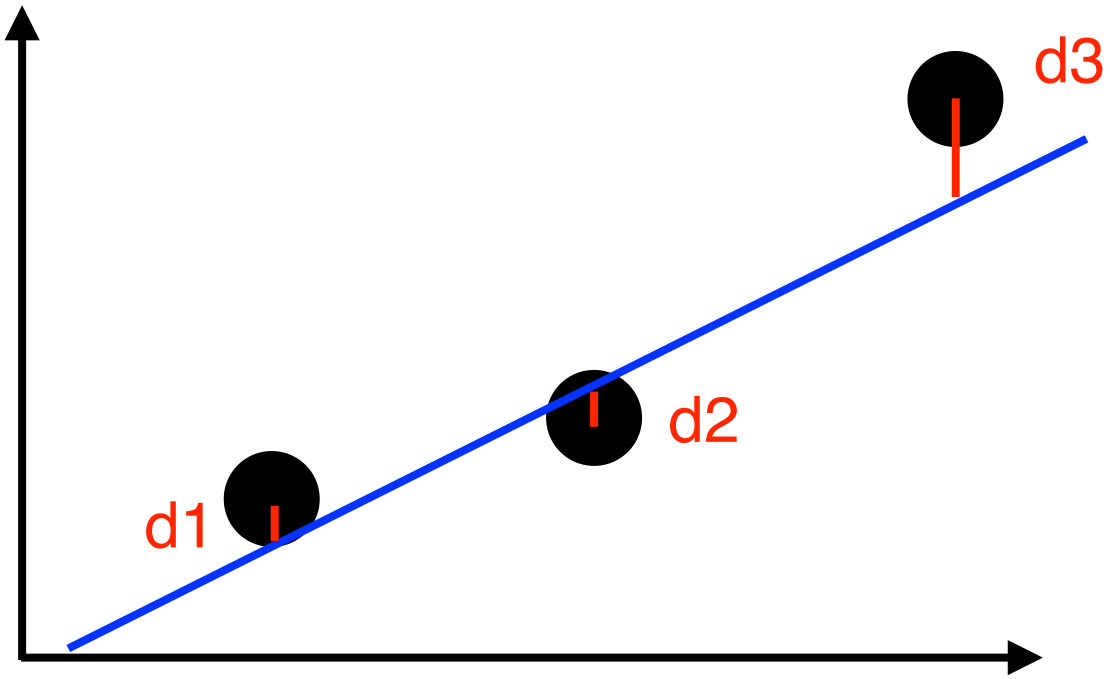
This looks like a reasonable guess and a good fit (of course, we will be more mathematically rigorous shortly)

How to do our linear fits?



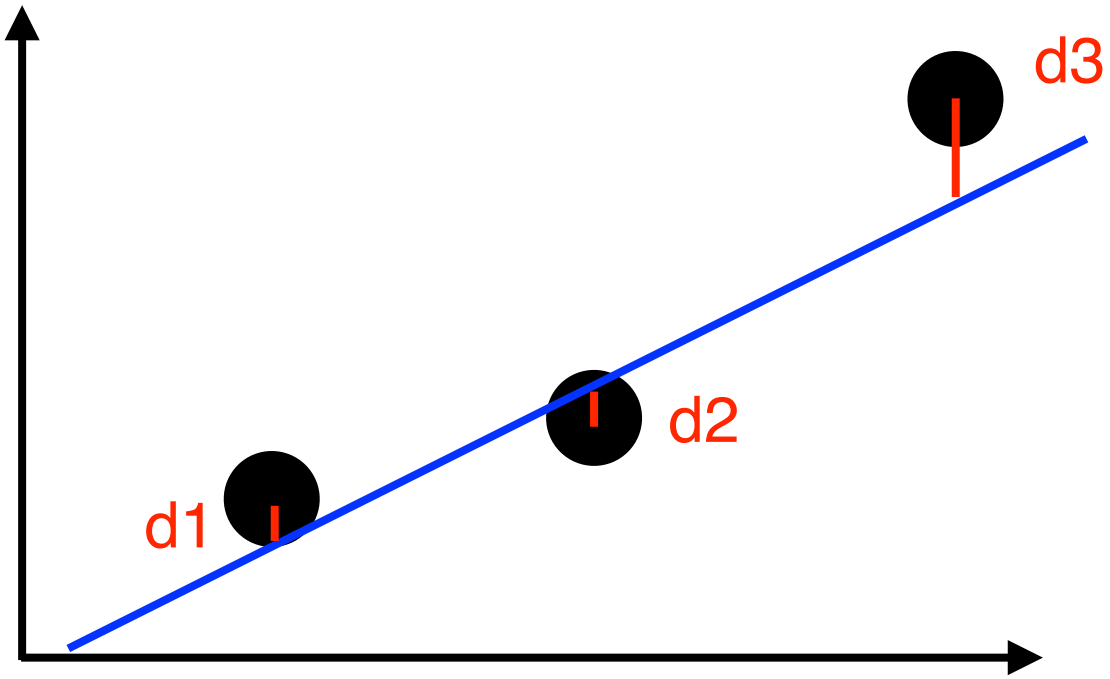
This is clearly not a good fit!

Why does the first line/fit look better?



Assume the errors are all the same (this assumption will be relaxed). Our straight line is given by $y(x) = a+bx$, so that our prediction for point x_i is $y_{pred,i} = a+bx_i$ and then the distance between this and the measured value y_i is given by $d_i = y_i - y_{pred,i} = y_i - a- bx_i$

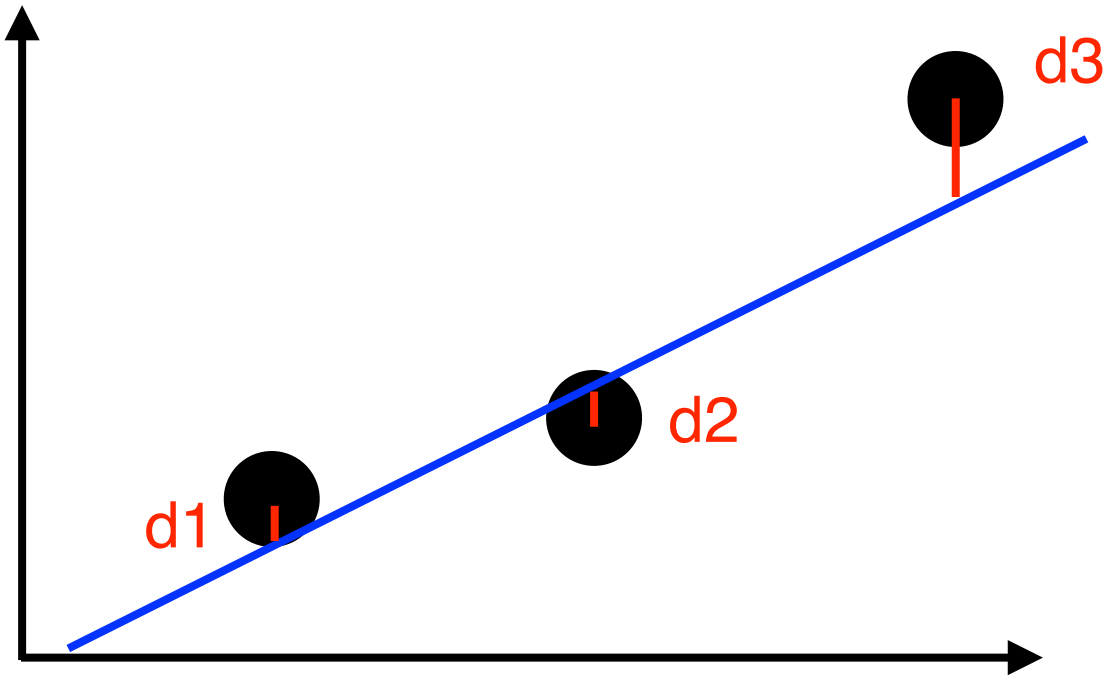
Why does the first line/fit look better?



The distance between the observed and measured value for point i is y_i - a- bx_i. Our total "distance" squared is then:

$$f(a,b) = \sum_i (y_i - a - bx_i)^2$$

How to get best-fit parameters



We want to **minimize** the following quantity:

$$f(a,b) = \sum_i (y_i - a - bx_i)^2$$

How to get best-fit parameters

We want to **minimize** the following quantity:

$$f(a,b) = \sum_i (y_i - a - bx_i)^2$$

$$\frac{\partial f}{\partial a} = -2\sum_i (y_i - a - bx_i)$$

$$\frac{\partial f}{\partial b} = -2\sum_i x_i(y_i - a - bx_i)$$

Two equations, two unknowns (a,b)

How to get best-fit parameters

$$\frac{\partial f}{\partial a} = -2\sum_i (y_i - a - bx_i) = 0$$

$$\frac{\partial f}{\partial b} = -2\sum_i x_i(y_i - a - bx_i) = 0$$

Sum over i takes N possible values

$$\sum_i (y_i - a - bx_i) = 0$$

$$\sum_i x_i(y_i - a - bx_i) = 0$$

$$\sum_i y_i - Na - b\sum x_i = 0$$

$$\sum_i x_i y_i - a\sum x_i - b\sum x_i^2 = 0$$

How to get best-fit parameters

$$\sum_i y_i - Na - b\sum x_i = 0$$

$$\sum_i x_i y_i - a\sum x_i - b\sum x_i^2 = 0$$

$$a = \frac{1}{N}\sum y_i - b\frac{1}{N}\sum x_i$$

$$a = \frac{1}{\sum x_i}\sum x_i y_i - b\frac{1}{\sum x_i}\sum x_i^2$$

How to get best-fit parameters

$$a = \frac{1}{N}\sum y_i - b\frac{1}{N}\sum x_i$$

$$a = \frac{1}{\sum x_i}\sum x_i y_i - b\frac{1}{\sum x_i}\sum x_i^2$$

Introduce notation to make this easier, set the 'a' values equal to each other

$$s_y/N - (b/N)s_x = (s_{xy}/s_x) - b(s_{xx}/s_x)$$

$$b(s_{xx}/s_x - s_x/N) = s_{xy}/s_x - s_y/N$$

$$b(Ns_{xx} - s_x^2) = Ns_{xy} - s_x s_y$$

$$b = (Ns_{xy} - s_x s_y)/(Ns_{xx} - s_x * s_x)$$

How to get best-fit parameters

$$a = \frac{1}{N}\sum y_i - b\frac{1}{N}\sum x_i$$

$$a = \frac{1}{\sum x_i}\sum x_i y_i - b\frac{1}{\sum x_i}\sum x_i^2$$

$$s_y/N - a = (b/N)s_x$$

$$s_{xy}/s_x - a = (b/s_x)s_{xx}$$

$$b = s_y/s_x - aN/s_x = s_{xy}/s_{xx} - as_x/s_{xx}$$

$$s_y/s_x - s_{xy}/s_{xx} = a(N/s_x - s_x/s_{xx})$$

$$a = (s_y/s_x - s_{xy}/s_{xx})/(N/s_x - s_x/s_{xx})$$

$$a = (s_y * s_{xx} - s_{xy} * s_x)/(N * s_{xx} - s_x^2)$$

How to get best-fit parameters

$$a = (s_y * s_{xx} - s_{xy} * s_x)/(N * s_{xx} - s_x^2)$$

$$b = (N * s_{xy} - s_x * s_y)/(N * s_{xx} - s_x^2)$$

Just a few simple sums to
calculate, pretty straightforward!

Is that the full picture?

We also want to know three additional, very important quantities:

1) The uncertainty on a
2) The uncertainty on b
3) The uncertainty per degree of freedom of the fit (aka the goodness of the fit). For our linear fit we have two constraints so the variance of the fit, or goodness of fit, is:

$$\sigma^2 = \frac{1}{n-2}\sum (y_i - y(x_i))^2$$

And what do we do if the uncertainty on each point is not equal? Minimize the chi2 instead!

$$\chi^2(a,b) = \sum \left(\frac{y_i - a - bx_i}{\sigma_i} \right)^2$$

Using point-by-point errors (if errors on each value are not all equal)

Uncertainties on parameters!

$$b = \frac{1}{S_{tt}} \sum_{i=0}^{n-1} \frac{t_i y_i}{\sigma_i} \,, \qquad a = \frac{S_y - S_x b}{S} \qquad \sigma_a^2 = \frac{1}{S}\left(1 + \frac{S_x^2}{S S_{tt}}\right) \,, \qquad \sigma_b^2 = \frac{1}{S_{tt}}$$

with

$$t_i = \frac{1}{\sigma_i}\left(x_i - \frac{S_x}{S}\right) \,, \qquad S_{tt} = \sum_{i=0}^{n-1} t_i^2 \,,$$

$$S = \sum_{i=0}^{n-1} \frac{1}{\sigma_i^2} \,, \qquad S_x = \sum_{i=0}^{n-1} \frac{x_i}{\sigma_i^2} \,, \qquad S_y = \sum_{i=0}^{n-1} \frac{y_i}{\sigma_i^2}$$

And goodness of fit :

$$\chi^2(a,b)/ndof = \frac{1}{n-2} \sum \left(\frac{y_i - a - b x_i}{\sigma_i} \right)^2$$

Using point-by-point errors

And goodness of fit :

$$\chi^2(a,b)/ndof = \frac{1}{n-2}\sum\left(\frac{y_i - a - bx_i}{\sigma_i}\right)^2$$

If our data really follow a linear distribution AND we have estimated our errors correctly, we should get chi2/ndf values ~1. Either way, we can use the chi2 and n to calculate a p-value indicating our goodness of fit

HW assignment!

In the git repo there is a folder with data.txt. Each line is a new set of data. The format is:

x,y,sigma_y,last

Where sigma_y is the estimated uncertainty on that y value and "last" tells you if this was the last input for this event (there should be 6 events in the file).

Write HLS code to do linear fits to data in each event. Note that you don't know in advance how big each event is, but you can assume a maximum of 250 events. Anything beyond that can be ignored. Try and optimize the code as much as you can.

For each event, return a,b, the uncertainty on those two parameters, the chi2/ndf and if you had to skip any events. Note that some of the events should fit very well to the model. Others might not!