Fitting Straight Lines to Data

Ian Hunter, MIT, 1 March 2018

We start by generating some Gaussian white noise which will serve as the measurement noise (error) added to the output.

$n := 201 \qquad i := 1 .. n$

$$e_i := -6 + \sum_{j=1}^{12} \mathrm{rnd}(1)$$

a convenient way to generate Gaussian white noise is to add together 12 uniformly distributed random numbers (such as are available in all computer languages) for each sample.

$$\mu_e := \frac{1}{n} \cdot \sum_{i=1}^{n} e_i \qquad \mu_e = 0.04$$

we check that the mean is near zero

$$\sigma_e := \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(e_i - \mu_e\right)^2} \qquad \sigma_e = 1.042$$

and note that the standard deviation is near one.

We actually want the mean to be zero and the standard deviation to be 50. This is achieved by setting

$$e_i := 50 \cdot \frac{e_i - \mu_e}{\sigma_e}$$

and check

$$\mu_e := \frac{1}{n} \cdot \sum_{i=1}^{n} e_i \qquad \mu_e = 0$$

$$\sigma_e := \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(e_i - \mu_e\right)^2} \qquad \sigma_e = 50$$

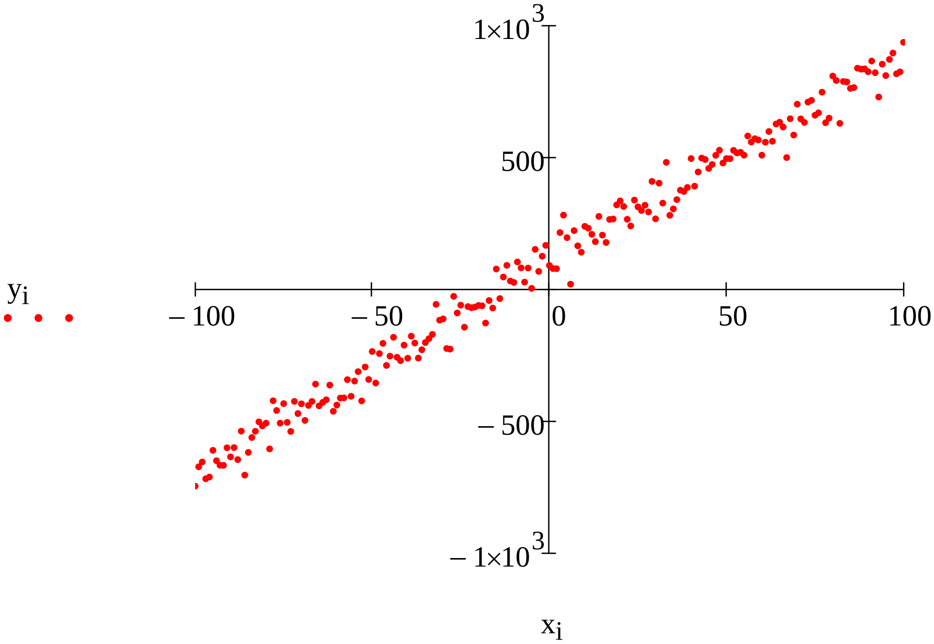
We next generate the input (numbers from -100 to 100)

$$x_i := i - 101$$

and then we generate the output, y_i , as a linear function (with specified slope, α and intercept, β) of the input, x_i , corrupted by additive noise, e_i

Let $\alpha := 8$ and $\beta := 100$

$$y_i := \alpha \cdot x_i + \beta + e_i$$



We now consider the problem of fitting a straight line to these data. Remember that a straight line is of the form $y_i = \alpha\, x_i + \beta$ where the slope is α and the intercept is β. Therefor when we say that we are fitting a straight line to our data that is equivalent to estimating α and β.

One approach is to fit by eye. A better approach is to define a criterion by which the "goodness" of the fit may be determined. A common criterion is the "least squares" or "least sum of squares" measure which simply sums the squares of the prediction errors. The RMS error ("root mean square" error) criterion achieves the same results (see below). The prediction errors are simply the vertical difference between the y values and the predicted y values (given by $\alpha\, x_i + \beta$). The parameters (α and β) which minimize the criterion are termed the optimal estimated parameters.

The predicted y values are

$$yp_i := \alpha \cdot x_i + \beta$$

The prediction errors are given by

$$pe_i := yp_i - y_i$$

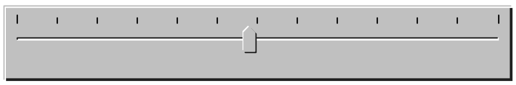
and the "sum of squares" and RMS criteria are

$$SS := \sum_{i=1}^{n} (pe_i)^2 \qquad RMS := \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (pe_i)^2}$$

Now try changing the slope and intercept until the RMS error is minimized (if you have a wheel on your mouse try using it to move the sliders).

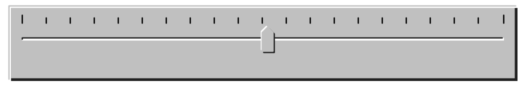
Slope

a1 :=



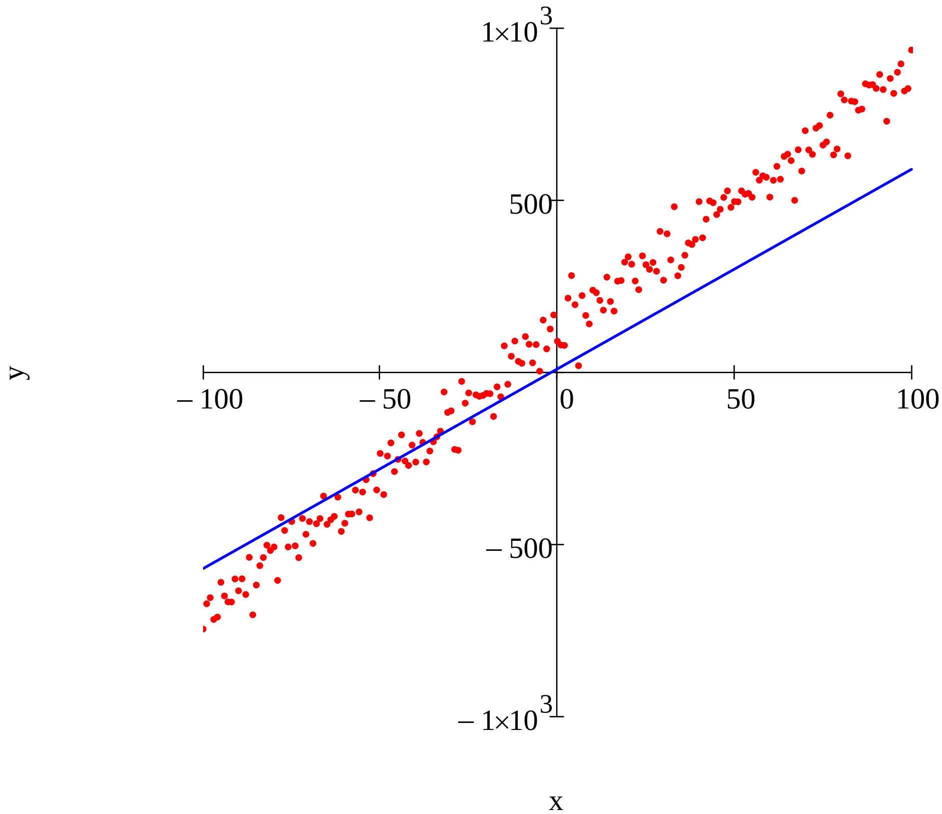
Intercep

a0 :=

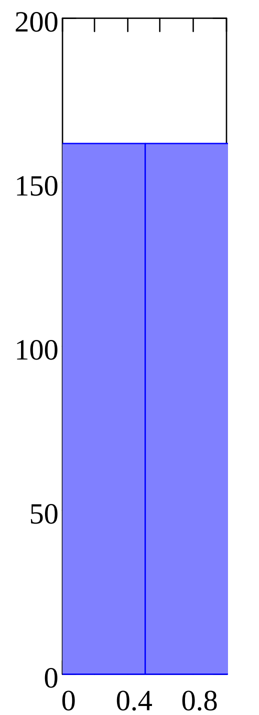


$\alpha = 8$

$\beta = 100$



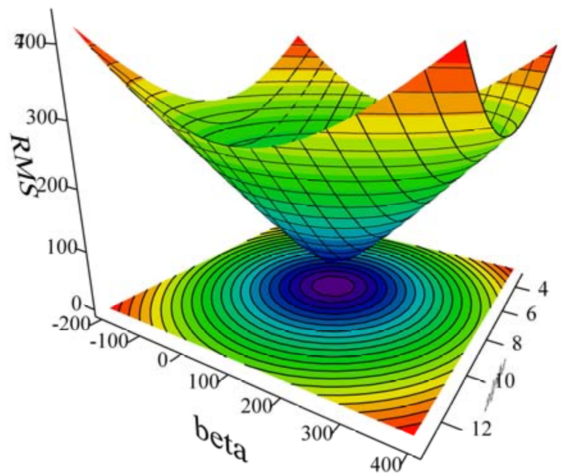
RMS = 161.625



We can see that the RMS depends on the values of the parameters a1 and a0 . We can therefore define the RMS as a function of a1 and a0 . Such a function is called an o**bjective function**.

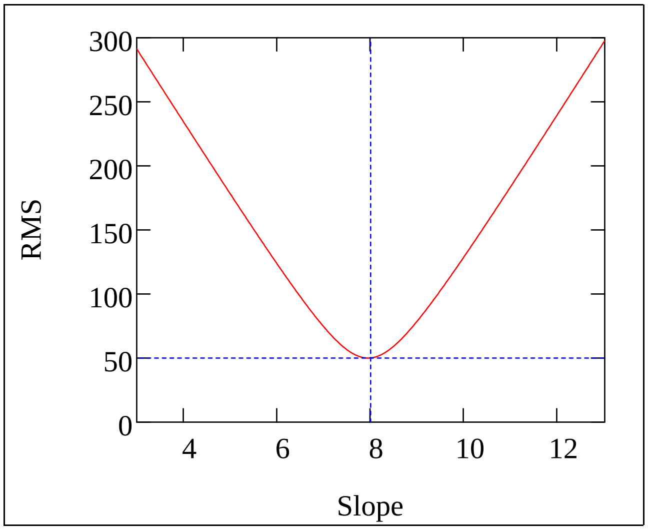
$$RMS(\alpha, \beta) := \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (\alpha \cdot x_i + \beta - y_i)^2}$$

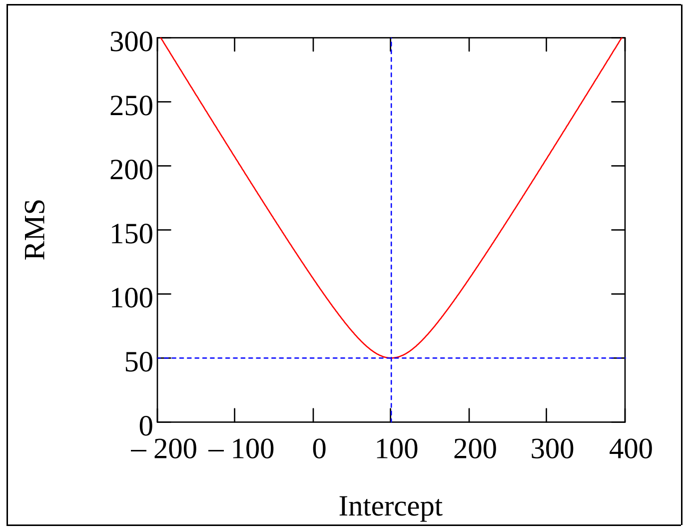
The RMS objective function is shown below (both as a surface plot and a contour plot). Use the mouse to rotate the plot about to see the form of this function.



RMS, RMS

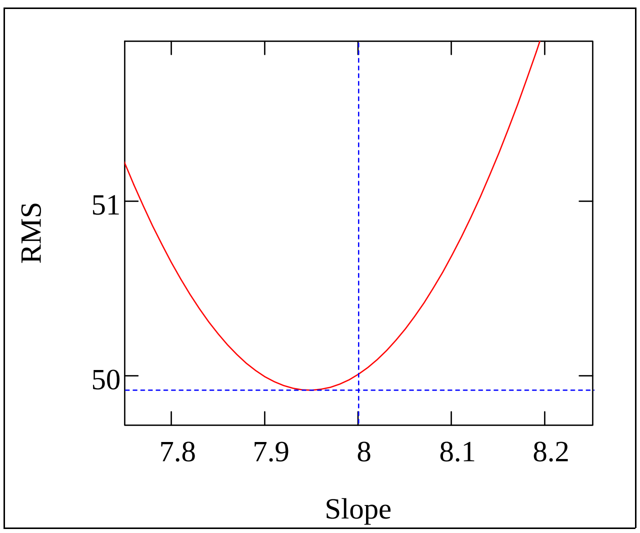
If we hold α constant (at its optimal value) and vary β we can see the way in which the RMS objective function changes or is "sensitive" to changes in β . We can do the same for the other parameter.

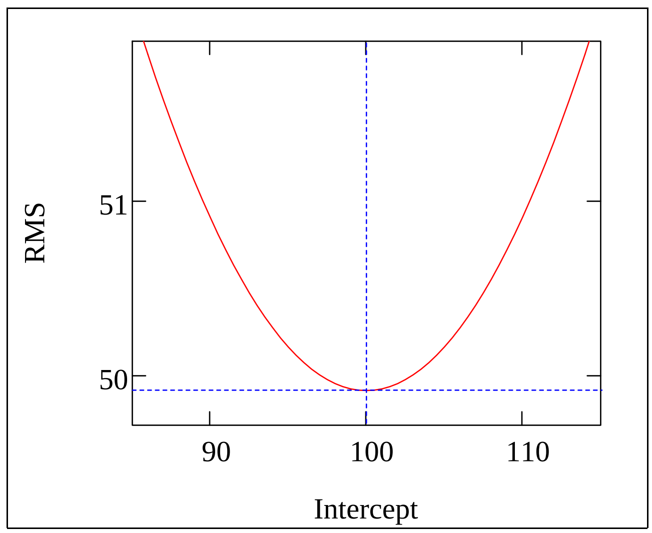




The objective function has two parameters (in this case). The one dimensional plots above constitute "slices" through the two dimensional RMS objective function and are called **sensitivity functions**.

We now zoom in on the sensitivity function minimums.





Notice that the "true" (original) parameter values are not always the same as those estimated by minimizing the objective function (or in this case the sensitivity functions).

So far we have determined the "best" parameter estimates by attempting to find the objective function (RMS) minimum by manually adjusting the parameters. What we need is an automatic way to find the objective function minimum.

Issac Newton proposed a fairly general technique to find the minimum of a function. **Newton's method** involves starting with some guesses as to the parameter values. These initial guesses are then successively improved by the following simple algorithm (where $a0_{new}$ and $a1_{new}$ are the current parameter estimates and $a0_{old}$ and $a1_{old}$ are the previous parameter estimates).

Minimization techniques (such as Newton's method, quasi-Newton techniques, conjugate gradient method, the method of steepest descent, the Levenberg-Marquardt technique) may be used to find the minimum (lowest point) of the objective function (and hence the "best fit" or optimal parameter estimates). These techniques start with some initial parameter estimates (i.e. a point on the objective function) . These parameter values are then adjusted in an attempt to follow a path down the surface torward the lowest point (or some sufficiently low point). The parameters at the lowest point are termed the "best fit" or optimal parameters. The process of successively trying new (and better) parameter values is called **iteration.**

However in the case of fitting some simple functions such as the straight line (as above) we do not need to use an iterative minimization technique to find the objective function minimum. Instead the optimal parameters, can be found directly (analytically).

$$\mathrm{RMS}(\alpha,\beta) := \sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}\left(\alpha\cdot x_i+\beta-y_i\right)^2}$$

The minimum of the RMS objective function corresponds to the point where the double derviative is zero. The minimum of RMS^2 occurs at the same parameter values as the minimum of RMS. Thus

We will save the original x and y and α and β values as

$xx := x \qquad yy := y \qquad alpha := \alpha \qquad beta := \beta \qquad nn := n$

$x := x \qquad y := y \qquad \alpha := \alpha \qquad \beta := \beta \qquad n := n$

The partial derivatives of RMS squared with respect to the parameters are

$$\frac{d}{d\alpha}\sum_{i=1}^{n}\left(\alpha\cdot x_i+\beta-y_i\right)^2 \rightarrow 2\cdot\beta\cdot\sum_{i=1}^{n}x_i - 2\cdot\sum_{i=1}^{n}\left(x_i\cdot y_i\right) + 2\cdot\alpha\cdot\sum_{i=1}^{n}\left(x_i\right)^2$$

$$\frac{d}{d\beta}\sum_{i=1}^{n}\left(\alpha\cdot x_i+\beta-y_i\right)^2 \rightarrow 2\cdot\alpha\cdot\sum_{i=1}^{n}x_i - 2\cdot\sum_{i=1}^{n}y_i + 2\cdot\beta\cdot n$$

These partial derivatives will equal zero at the minimum of the objective function (i.e. slopes are zero at this minimum).

We have two unknowns (α and β) and two independent equations so we can solve for α and β

$x := xx \qquad y := yy \qquad n := nn$

$$\alpha := \frac{n\cdot\sum_{i=1}^{n}(x_i\cdot y_i)-\sum_{i=1}^{n}x_i\cdot\sum_{i=1}^{n}y_i}{n\cdot\sum_{i=1}^{n}(x_i\cdot x_i)-\sum_{i=1}^{n}x_i\cdot\sum_{i=1}^{n}x_i} \qquad \alpha = 7.948$$

$$\beta := \frac{\sum_{i=1}^{n}(x_i\cdot y_i)\cdot\sum_{i=1}^{n}x_i-\sum_{i=1}^{n}(x_i\cdot x_i)\cdot\sum_{i=1}^{n}y_i}{\sum_{i=1}^{n}x_i\cdot\sum_{i=1}^{n}x_i-n\cdot\sum_{i=1}^{n}(x_i\cdot x_i)} \qquad \beta = 100$$

Note that these formulas are notoriously sensitive to numerical round-off errors. More accurate numerical methods are available.