Fitting Functions to Data

Ian Hunter, MIT, 1 March 2018

Power Functions

To illustrate the process of fitting a function to data we will consider the case of a power function of the form

$$y = \alpha\, x^{\beta}$$

where x is the independent variable (i.e. experimentally controlled variable or input) and y is the resulting dependent variable (or output). The parameters α and β must be determined by fitting the power function to the x,y data set. This process is called **parameter estimation**.

In order to make the problem realistic we will add some noise to y to simulate noisy measurements. Measurement noise is often (though not always) found to be independent (i.e. uncorrelated) and via analogy to white light is usually called "white" because it has a flat power spectrum (all frequencies present have equal amplitude). Measurement noise is often (though not always) found to have a Gaussian (i.e. normal) probability density. Such noise is therefore called Gaussian white noise. Much of signal and system analysis and parameter estimation assumes that zero mean, constant standard deviation, Gaussian white noise is added to the output.

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

$n := 100 \qquad i := 1 .. n$

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

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.

$$\mu_e := \frac{1}{n}\cdot\sum_{i=1}^{n} e_i \qquad \mu_e = 0.02$$

we check that the mean is near zero

$$\sigma_e := \sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}\left(e_i - \mu_e\right)^2} \qquad \sigma_e = 1.059$$

and note that the standard deviation is near one.

We actually want the mean to be zero and the standard deviation to be 10. This is achieved by setting

$$e_i := 10 \cdot \frac{e_i - \mu_e}{\sigma_e}$$

and check

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

$$\sigma_e := \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(e_i - \mu_e\right)^2} \qquad \sigma_e = 10$$

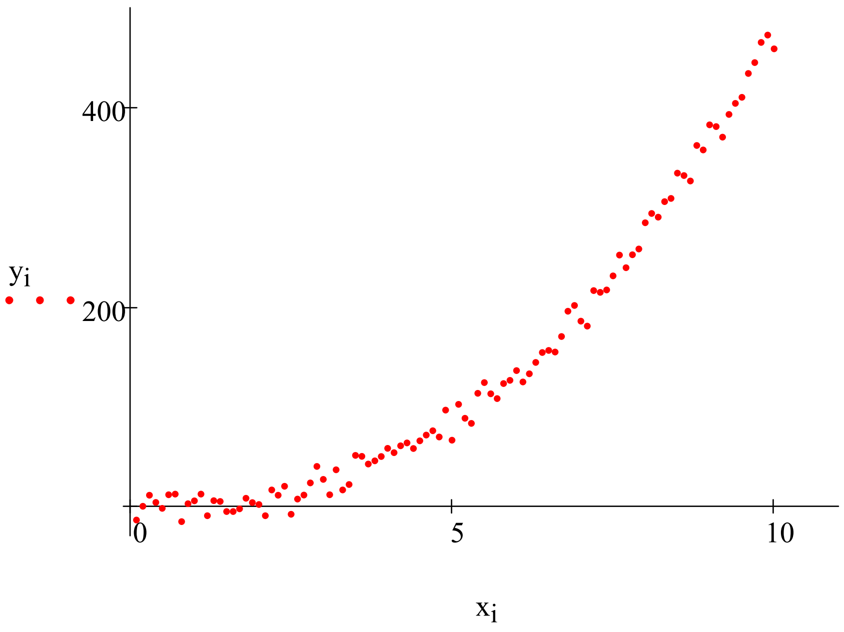
We next generate the input (numbers from 0 to 10)

$$x_i := i \cdot 0.1$$

and then we generate the output, y_i , as a power function (with specified parameters α and β) of the input, x_i , corrupted by additive Gaussian white noise, e_i

Let $\alpha := 1.5$ and $\beta := 2.5$

$$y_i := \alpha\cdot\left(x_i\right)^{\beta} + e_i$$



We now pretend that we do not know what the parameters α and β were and consider the problem of fitting a power function to these data. Remember that a power function is of the form $y_i = \alpha\, x_i^{\ \beta}$. Therefore when we say that we are fitting a power function to our data that is equivalent to estimating the parameters 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_i values (given by $\alpha\, x_i^{\ \beta}$). The parameters (α and β) which minimize the criterion are termed the optimal estimated parameters (often called the "best fit" parameters).

The predicted y values are

$$yy_i := \alpha\cdot\left(x_i\right)^{\beta}$$

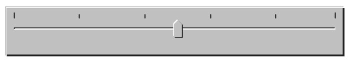
The prediction errors are given by

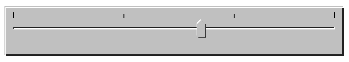
$$ee_i := yy_i - y_i$$

and the "sum of squares" and RMS criteria are

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

Now try manually changing the α and β parameter values until the RMS error is minimized (if you have a wheel on your mouse try using it to move the sliders).

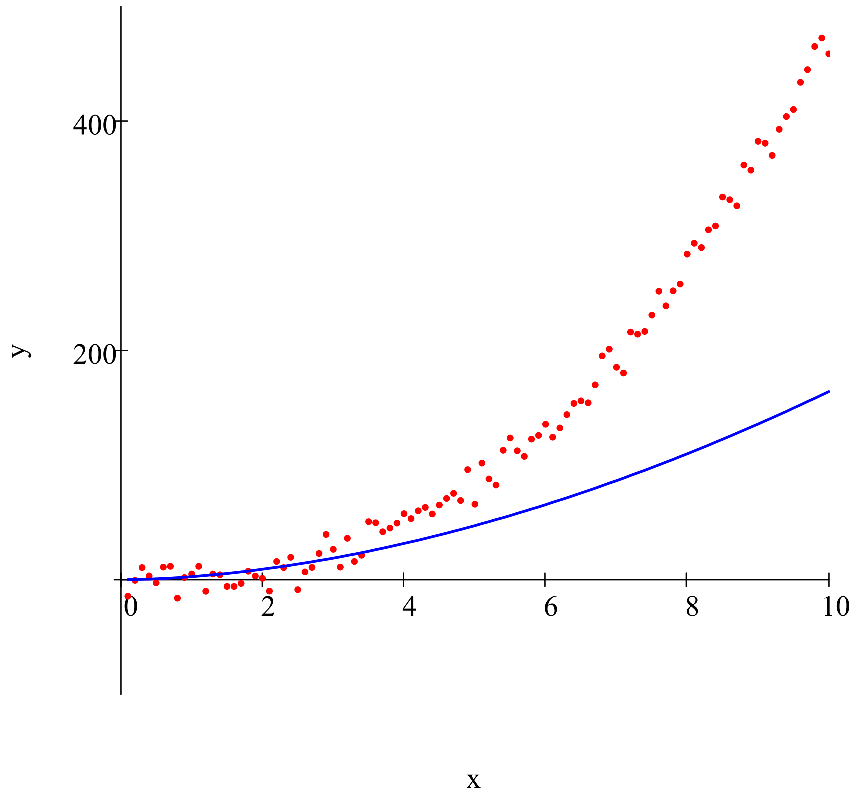
$\alpha :=$ 

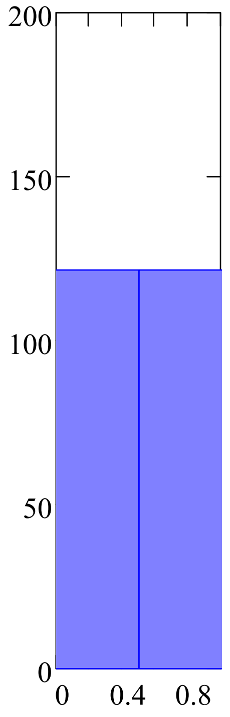
$\beta :=$ 

$\alpha = 2.6$

$\beta = 1.8$

RMS = 121.683

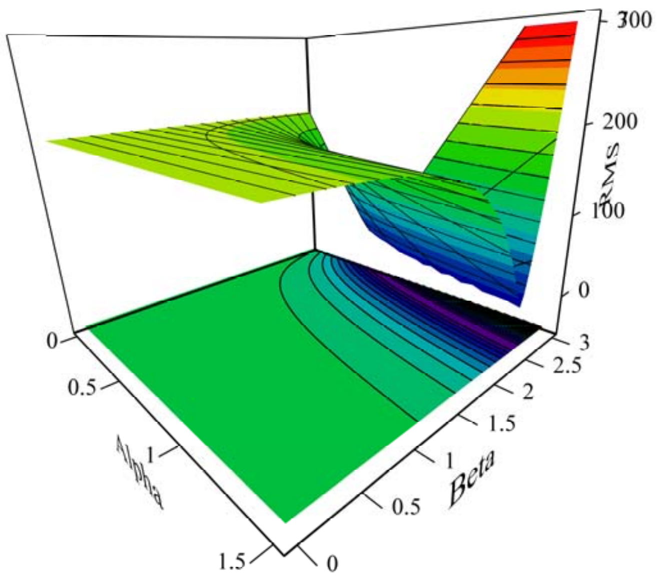




We can see that the RMS depends on the values of the parameters α and β . We can therefore define the RMS as a function of α and β . Such a function is called an **objective function**.

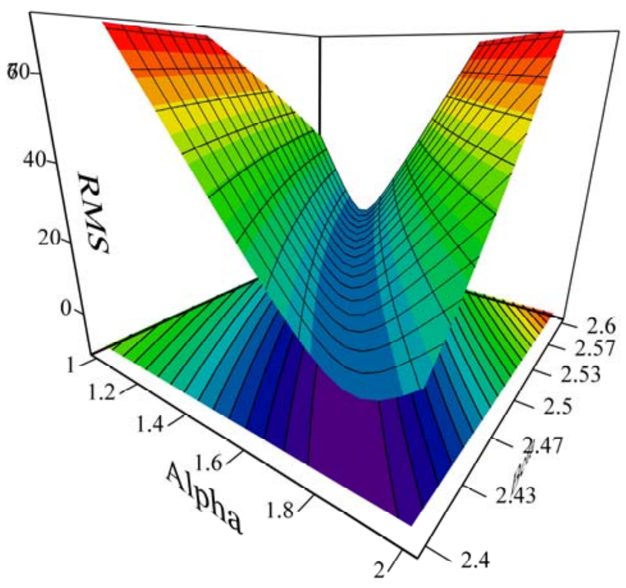
$$\mathrm{RMS}(\alpha,\beta) := \sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n}\left[\alpha\cdot\left(x_i\right)^{\beta} - y_i\right]^2}$$

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.



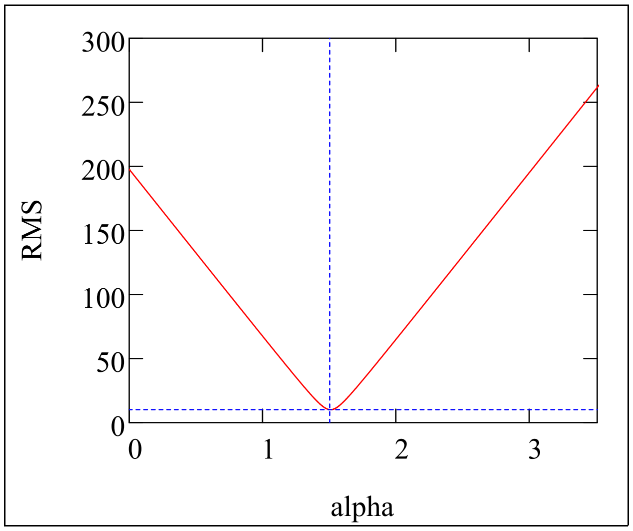
RMS, RMS

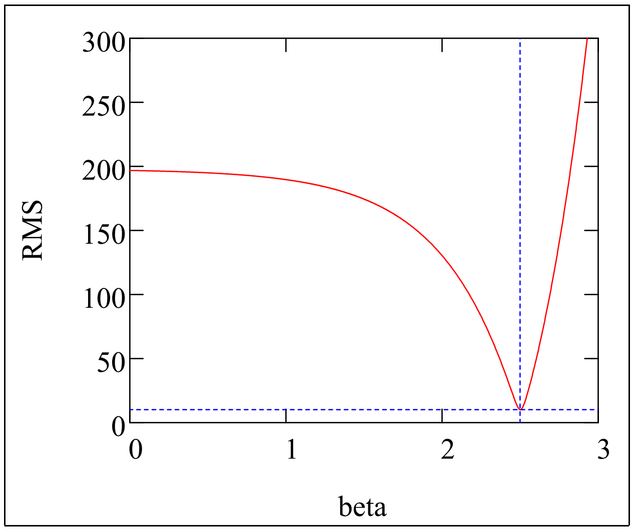
Lets zoom in on the region about the minimum.



RMS, RMS

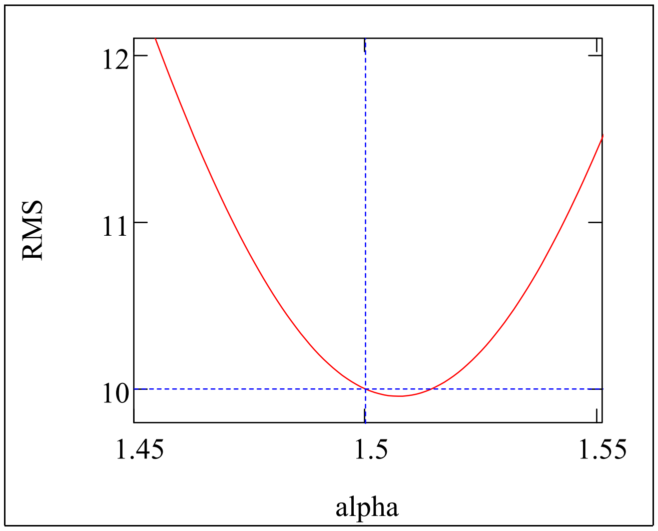
If we hold α constant (at its optimal value) and vary β we can see the way in which the RMS objective function changes or is "sensitive" to changes in β . We can do the same for the other parameter.

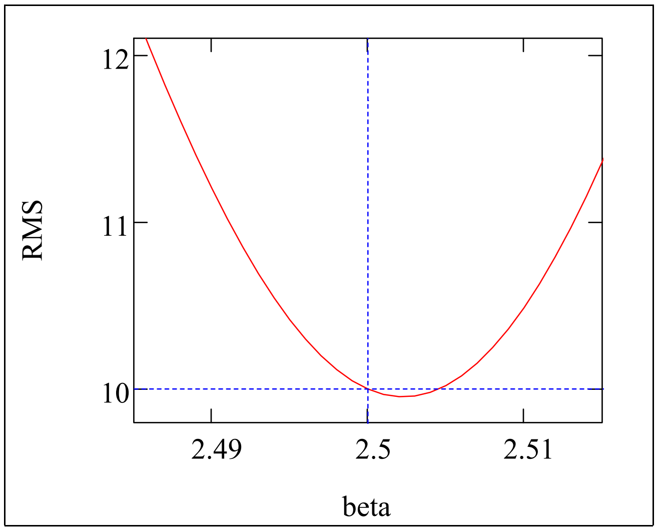




The objective function has two parameters (in this case). The one dimensional plots above constitute "slices" through the two dimensional RMS objective function and are called **sensitivity functions**.

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). Note also that the objective function behaves like a quadratic (i.e. second-order polynomial) about the minimum. This property is used by iterative minimization techniques discussed briely below.

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.

Analytic Solutions

Certain problems, such as fitting a linear function of the form $y = \alpha\, x + \beta$ to data using a least squares objective function, turn out to have an analytic (closed form, non-iterative) solution from which the parameter estimates can be immediately found. There is no such analytic solution to the power function parameter estimation as detailed above.

Linearization via Transformation

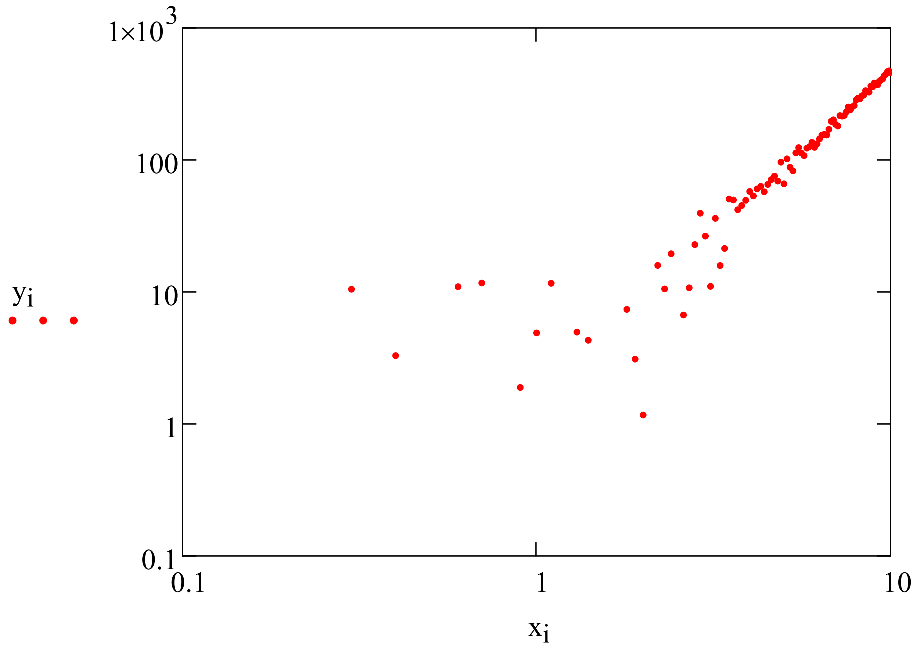
For other problems a transformation applied to either or both the x and y variables will sometimes "linearize" the problem so that a linear function may be fitted analytically. For example the power function above may be "linearized" by log transforming both axes. A power function on log log co-ordinates will be a straight line

$$\log(y) = m \log(x) + c$$

from which the original α and β parameters may be estimated from the slope, m, and intercept, c via

$\alpha = 10^{c}$ and $\beta = m$

Let's look at the data on a log log plot.



The problem with such transformations is that they sometimes (as in this case) linearize the function but in doing so distort the noise probability densities (which in this case are no longer Gaussian). A least squares line fitted to this log log plot will not give the same parameter estimates as the original nonlinear power function fitted to the untransformed data. Another problem is that the log of the negative values of y are undefined and so cannot be plotted on the log log plot and cannot contribute to the parameter estimates.

Iterative Minimization Techniques

In general closed form solutions do not exist and an iterative ("trial and error') minimization technique must be used.

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). As before 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.**

Mathcad provides a fairly powerful set of minimzation techniques from which it will automatically select to solve a minimization problem.

We start by defining a suitable objective function (usually it will be least squares unless the noise is non Gaussian)

$$SS(\alpha,\beta) := \frac{1}{n}\cdot\sum_{i=1}^{n}\left[\alpha\cdot\left(x_i\right)^{\beta} - y_i\right]^2$$

We then provide some inital parameter estimates

$\alpha\alpha := 1$ $\beta\beta := 1$

Given

$$\begin{pmatrix}\alpha\alpha\\ \beta\beta\end{pmatrix} := \text{Minimize}(SS, \alpha\alpha, \beta\beta)$$

this implements the minimization technique

Our estimated parameters are

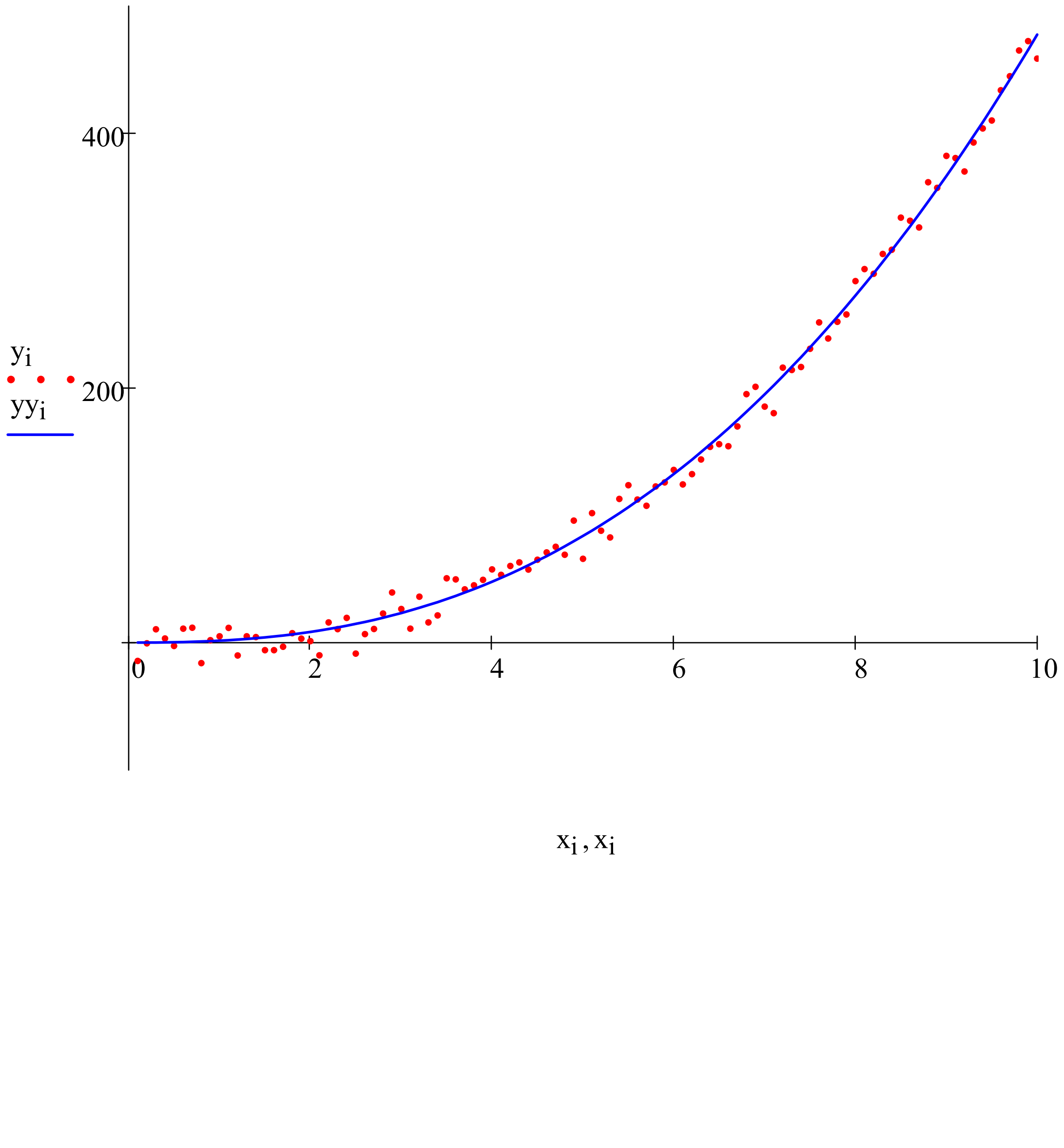
$\alpha\alpha = 1.471$ $\qquad\qquad$ $\beta\beta = 2.511$

These are close to the "true" values of

$\alpha = 1.5$ $\qquad\qquad$ $\beta = 2.5$

We can now generate the predicted yy using the estimated parameters

$$yy_i := \alpha\alpha\cdot\left(x_i\right)^{\beta\beta}$$


y_i
yy_i
x_i, x_i