Example Code

Suppose we are interested in finding the vertical intercept "a" of a dataset consisting of three variables: y is a function of two independent variables x_1 and x_2 with the dependency:

$$y(x_1, x_2) = a + b{\cdot}x_1 + c{\cdot}x_2^2$$

At each measurement of $y(x_1, x_2)$ there is random noise present. The parameters (a, b, c) are estimated with a multivariate fit to the data which is averaged over N measurements of $y(x_1, x_2)$. The quality of the fitted parameter a is evaluated as a function of N to test how many measurements are required to obtain a good estimate.

For example, this could represent the following investigation. We believe from an analytical model that the intensity of light received at a pixel located at (x_1, x_2) is given by

$$y(x_1, x_2) = a + b{\cdot}x_1 + c{\cdot}x_2^2$$

when observing a feature on the surface of a star. There is also a random level of noise present at each pixel associated with stochastic effects arising from the photosphere of the star. In order to estimate the value a of the light intensity present at $(0, 0)$ on the CCD we take several photographs. An estimate of (a, b, c) is made for each photograph based on a fit to the function $y(x_1, x_2)$ and then an overall estimate for a is found by averaging the value calculated from each photograph.

The following code implements this idea of testing how many repeated measurements are required to produce a high quality multivariate fit in the presence of random noise. **The code is interactive and therefore should be run from the command-line** in order to test the following three features it contains:

1. Plotting a single multivariate fit to the data in 2D
2. Plotting a single multivariate fit to the data in 3D
3. Running a user-input selected set of simulations and plotting the fitted parameter as a function of the number of trials in each simulation