How to do a linear analysis of data:

The first step is to see if indeed this process makes any sense at all. That is, do the data appear to be linearly related? This step is usually accomplished by doing a "scatter plot" of the data. This means plotting them as x-y points on a graph. The choice of ranges for the axes matters here a great deal. A poor choice for the range of values on one of the axes could lead to all the points falling over a very small portion of the graph, making it impossible to tell whether there is any potential relationship in the data set. Thus, choose your axes so that the range of them is approximately the same as that in your data. So if you have a set of body weights ranging from 5 grams to 160 grams, it would be reasonable to choose the axis for body weight to run from 0 – 160 grams or so.

Which is the x (horizontal) axis and which is the y (vertical)? If you have some reason to expect that one of the measurements is the dependent one (e.g. body weight depends on age, rather than the reverse), then choose the one that is dependent (weight) as the y axis and the independent one (age) as the horizontal axis. If you do not have any reason to expect that one of the measurements is caused by the other, then the choice of which measurement is plotted on which axis doesn't matter. Just choose one and go with it.

The next step is to eyeball the data and see if there appears to be any relationship. If the scatter plot looks like the points might be described at least approximately by a line (e.g. don't worry if there is a lot of scatter about any line you might draw on the graph), then it is reasonable to proceed with fitting a line. The next step then is eyeballing the data and making a guess at a line without doing any calculations or using any program. This will not give you an exact answer, but it will be useful later in checking to see that the line that is obtained from a computer or other method you choose makes sense. To eyeball, just quickly draw a line through the data, eyeball a rough slope and then write down the formula for the line using the point-slope form, that is $y - y_1 = m * (x - x_1)$ where m is the slope and the point you have chosen is (x_1,y_1), or use the point-intersept form if it can easily be estimated where the line crosses the y-axis $y = m * x + b$ where b is the y-intersept. Be sure in doing this that you keep your units straight, so that you know what units each measurement, and thus the slope, is in.

What if the data do not appear to be linearly related? Then don't try to fit a straight line. Later we'll discuss ways to transform the data to see if a different relationship might be a better choice.

If it looks like a linear relationship is a reasonable assumption, the next step is to find the "least squares fit". This can be done automatically within Matlab, and many calculators can also do this. The idea is to choose a line that "best fits" the data in that the data points have the minimum sum of their vertical distances from this particular line. One way that you might code a computer to do this is:

(i) Guess at the equation for the line - y = a*x + b.
(ii) Measure the vertical distance from each data point to the line you have chosen.
(iii) Sum up the distances chosen in (ii).
(iv) Change the slope a and the intercept b to see if you can reduce the sum obtained in (iii).
(v) Continue this until you get tired or you've done the best you can.

One of the potential problems with the above is what "distance" to use in (ii). If you allow some distances to be positive and some negative (e.g. for a point above the line and a point below the line), then the distances can cancel, which is not appropriate. So we want all distances to be positive and the standard way to do this is the just square each vertical distance found in (ii), and produce the sum of the square distances to get (iii). The line we would get by going through step (v) would then be called the "least squares fit" since it chooses the line so as to minimize the sum of the square deviations of points from the line.

It turns out that it is not necessary to go through the steps (1)-(v) above at all. It can be proven that the "best" values of the slope a and the intercept b can be obtained from a relatively simple formula that just uses the x and y values for all the data points. Matlab does this easily for you using the command "C=polyfit(A,W,1)" which will produce a vector C in which the first value is the best fit for the slope a and the second is the best fit for the intercept b for the least squares fit of the vector of data W (on the vertical axis) to the vector of data A (on the horizontal axis). Think of A as giving a vector of ages (in days) of bats and the vector W giving the weights of these bats (in grams). Note that the units of the components of C depend upon the units the data are measured in. The first component of C is a slope so it has units grams per day for the bat example, and the second component of C (the y-intercept) has the same units as the measurement on the y-axis (grams in the case of the bats).

Once you have a linear least squares fitted line, you can proceed to use it to interpolate (find the y-value predicted by the linear fit for an x-value that falls in the range of the x-values in your data set), or to extrapolate (find the y-value predicted by the linear fit for an x-value that falls outside the range of the x-values in your data set). Thus if you have values for body weights W (in grams) and ages of bats A (in days) 9, 15, 20, 22, 34, 44 and 49, and it appears that a linear fit to the data is reasonable, you can interpolate to find the weight of a bat of age 30 days, or you can extrapolate to find the weight of a bat of age 60 days. All you do to find these weights is to plug the age into the equation of the line and calculate the associated y-value (weight in grams). Matlab makes this easy by using the command "Y=polyval(C,30)" which will give the best guess according to the linear fit for the weight of a bat of age 30 days.

How can we tell if a linear fit is any good?

This is where we make use of the notion of a correlation. In common parlance, we say two measurements are "correlated" if there appears to be some relationship between them, though this relationship need not be causal. Thus leaf length and width might be related to each other, but neither is caused by the other. They might simply be related due to the age of the leaf or the environmental conditions under which the leaf developed (e.g. better nutrients and water could lead to a larger leaf).

There is a formal definition of correlation that we will use which essentially tells how close to linearly related two measurements are. Note that this is restricted to being a measure for linear relationships. If two measurements are related, but not linearly, then the correlation we estimate may not imply the measurements are closely related when they actually are. For example, human body weight is certainly related to age as an individual grows, but growth is not linear at all and so a correlation may not be the best way of saying that these two variables are related.

Correlation is measured by the "correlation coefficient" for which the small Greek letter rho (ρ) is typically used. The calculation of ρ follows easily from the x and y values of the data set. The coefficient ρ is a measure of the strength of the relationship between the two measurements, scaled in such a way that if two measurements fell exactly on a straight line with positive slope, then $\rho = 1$, while if they fell exactly on a line with negative slope, $\rho = -1$. In these cases we say the data are "perfectly positively correlated" or "perfectly negatively correlated". If $\rho = 0$ we say the data are "uncorrelated" but again this doesn't mean the data are not related - for example if the data fell on a parabola $y = (x-1)^2$ for values of x between 0 and 2, the correlation would be near zero but the data are certainly closely related.

Again Matlab makes it easy to calculate the correlation coefficient of two vectors using "corrcoef(A,W)" which computes the correlation coefficient of the data given in the two vectors A and W. Many calculators will compute this as well. Note that the correlation coefficient is a dimensionless number - in calculating it the dimensions of the measurements cancel out.