Error Analysis for Straight Line Fits

The following borrows liberally from *Numerical Recipes: The Art of Scientific Computing*. This classic book is available on line at http://www.nr.com. If you don't mind having the code in an obsolete language (like Fortran 77), you can click on "Obsolete Versions" and read the whole book for free. The part that explains the equations going into the code is the same (and what I used to generate this file). You can buy an online version with current code there, as well.

In general, all of the following is built in to standard data analysis programs, such as Excel, Igor, Matlab, Mathematica, Origin, etc. It is OK to use them, but you should read the manual to figure out exactly what they are doing. For one of your labs this quarter that involves a linear fit, it is probably worth doing this by hand once just to get a good feel for what happens behind the scenes in your canned fitting software.

Useful Statistical Parameters you can calculate from the data:

1. Mean: $\bar{y} = \langle y \rangle = \frac{1}{N}\sum_{i=1}^{N} y_i$
2. Variance: $Variance = \frac{1}{N-1}\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2$. If *N* is large (so *N* ~ *N-1*), this reduces to $\sigma^2 = \left\langle y_i^2 \right\rangle - \left\langle y_i \right\rangle^2$ where $\langle\ \rangle$ denotes the average.
3. Standard deviation: $\sigma = \sqrt{Variance}$
4. According to *Numerical Recipes*, if the variance is large, it is often more robust to estimate errors from the absolute value of the deviations rather than the square. This is especially true when you have one highly aberrant data point that gets very strongly weighted in the variance, and might well be aberrant for systematic rather than random reasons.

   $$Adev = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \bar{y}\right|$$

   If the data have some strange outliers, you might well want to substitute the median for the mean in this formula.

Assessing the fit between your data and a model:

The most commonly used method (least squares) is to optimize the model to minimize the value of Chi-squared: $\chi^2 \equiv \sum_{i=i}^{N}\left(\frac{y - y_i}{\sigma_i}\right)^2$, where σ_i is the standard deviation of the measurement of y_i, and $y = y\left(x \mid p_1, p_2, \ldots, p_M\right)$ is your model, where p_j are the various parameters in the model and *x* is the independent variable.

Note χ^2 will scale with the number of data points you have, and a general rule of thumb is that you have a reasonably good fit when $\chi^2 \sim N - M$ where *N* is the number of data points

and *M* is the number of fitting parameters. When this is true, deviations from the fit are comparable to the random error in the measurement. Frequently, people reduce χ^2 by this factor, $\chi^2_{red} = \frac{\chi^2}{N-M}$, and talk about having a good fit when $\chi^2_{red} \sim 1$. [N.B. the subscript is often missing, and people just call this reduced value χ^2). If χ^2_{red} is very different from 1 when you are done, then either your model is wrong or your estimate of σ is off.

Finding parameters in the fit by minimizing χ^2.

The "least-squares best fit" for a parameter is when χ^2 is minimized, or

$$0 = \sum \left(\frac{y - y_i}{\sigma_i^2} \right) \frac{\partial y(x_i \mid \{..., p_k, ...\})}{\partial p_k},$$

which yields a set of *M* simultaneous equations. This can be solved exactly if the parameters are linear and there is no uncertainty in the *x* measurements, but needs to be solved iteratively for non-linear models or if *x* and *y* both have variance.

Application to a linear fit:

For a linear fit, your model is $y = a + bx$. This leads to a χ^2:

$$\chi^2(a,b) = \sum_{i=1}^{N} \left(\frac{a + bx_i - y_i}{\sigma_i} \right)^2.$$

Taking the derivatives, and defining some useful sums:

$$0 = \frac{\partial \chi^2}{\partial a} = 2 \sum_{i=1}^{N} \frac{a + bx_i - y_i}{\sigma_i^2} = 2\left(aS + bS_x - S_y \right)$$

$$0 = \frac{\partial \chi^2}{\partial b} = 2 \sum_{i=1}^{N} \frac{x_i\left(a + bx_i - y_i\right)}{\sigma_i^2} = 2\left(aS_x + bS_{xx} - S_{xy} \right)$$

or:

$$aS + bS_x = S_y$$
$$aS_x + bS_{xx} = S_{xy}$$

where:

$$S = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \qquad S_x = \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2} \qquad S_y = \sum_{i=1}^{N} \frac{y_i}{\sigma_i^2}$$

$$S_{xx} = \sum_{i=1}^{N} \frac{x_i^2}{\sigma_i^2} \qquad S_{xy} = \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2} \qquad \Delta = SS_{xx} - \left(S_x\right)^2$$

Combining all of this we find: $\boxed{a = \frac{S_{xx}S_y - S_x S_{xy}}{\Delta} \qquad b = \frac{SS_{xy} - S_x S_y}{\Delta}.}$

where Δ is the determinant of coefficients, and also relates to both the variance of the independent variable distribution and the variance in the individual dependent variable measurements. If all the data have the same standard deviation σ, then this becomes $\Delta = \frac{N^2}{\sigma^4}\left[\left\langle x^2 \right\rangle - \left\langle x \right\rangle^2 \right]$, where $\left\langle f(x) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$.

Note that if σ is the same for each data point, the value of σ doesn't change the value of *a* or *b*, but it does impact the magnitude of χ^2 and your estimated uncertainty in those parameters. In a commercial program, if you don't input a specific uncertainty wave, it calculates it from your data assuming it is normally distributed around your model.

Uncertainties in fitted parameters:

If the variations are random and uncorrelated between data points (i.e., whether a particular data point is high or low is independent of the previous data point being high or low), then we can propagate the errors in y_I into the error in something that is a function of the y_i values, namely our fit parameter.

If the errors are uncorrelated, then they add quadratically, and the contributions are weighted by how much the parameter would vary if the data varied. In equations, for parameter p_j:

$\sigma_{p_j}^2 = \sum \sigma_i^2 \left(\frac{\partial p_j}{\partial y_i} \right)^2$. Here σ_i is the standard deviation of the i^{th} data point.

For the straight line, we can take derivatives of the equations on the previous page:

$$\frac{\partial a}{\partial y_i} = \frac{S_{xx} - S_x x_i}{\sigma_i^2 \Delta} \qquad \frac{\partial b}{\partial y_i} = \frac{S x_i - S_x}{\sigma_i^2 \Delta}$$

Squaring and performing the sums then yields:

$$\sigma_a^2 = \frac{S_{xx}}{\Delta} \ \textit{and} \ \ \sigma_b^2 = \frac{S}{\Delta}.$$

Having Δ in the denominator means when your *x* values have a small range relative to their mean, you have big error bars on your slope and intercept. Likewise, when your variance in your measurement is big, you have big error bars on your fitted parameters.

In general, there is a correlation amongst the uncertainties in multiple parameters. For a straight line, this "coefficient of correlation" for the intercept *a* and the slope *b* is:

$$r_{ab} = -\frac{S_x}{\sqrt{S S_{xx}}}.$$

If $r_{ab} > 0$, then the errors in *a* and *b* are likely to have the same sign, while if $r_{ab} < 0$, they are likely to be anti-correlated (if *a* is high, then *b* is low).

Errors in both x and y

First, we generate a new χ^2 function taking both error ranges into account. Errors in *x* yield an error in *y* that scales with the slope of the line, and if the two are otherwise uncorrelated they add in quadrature.

$$\chi^2 = \sum_{i=1}^{N} \frac{\left(a + bx_i - y_i\right)^2}{\sigma_{y,i}^2 + b^2\sigma_{x,i}^2}$$

Let the denominator term $\sigma_{y,i}^2 + b^2\sigma_{x,i}^2 = \frac{1}{w_i}$.

With *b* in both the numerator and denominator, we get a non-linear equation for *b* when we take the derivative, but we can solve for *a*:

$$\frac{\partial \chi^2}{\partial a} = 0 = 2\sum_{i=1}^{N} w_i\left(a + bx_i - y_i\right), \text{ or } a = \frac{\sum_{i=1}^{N} w_i(y_i - bx_i)}{\sum_{i=1}^{N} w_i}.$$

To fit for *a* and *b* simultaneously, it is best to scale the data so they each have similar magnitudes for their variances. Since we can solve for *a* given a value of *b*, we guess *b*, solve for *a* and then minimize χ^2 with respect to *b*, keeping on re-calculating *a* along the way.

One way to find the minimum with respect to *b* if you have a reasonable first guess is to calculate χ^2 for three values of *b* that you think bracket the right answer without having a maximum in between (varying *a* with each value of *b* to minimize χ^2 for that particular *b*), fit a parabola to these calculated χ^2 values, and then find the combination of *a* and *b* that gives a global minimum in χ^2.

If the three points are b_1, b_2, and b_3 (in rank order, with b_2 your initial guess for *b*), then the minimum of the parabola that goes through those three points is at:

$$b_{\min} = b_2 - \frac{1}{2}\frac{\left(b_2 - b_1\right)^2\left[\chi^2(b_2) - \chi^2(b_3)\right] - \left(b_2 - b_3\right)^2\left[\chi^2(b_2) - \chi^2(b_1)\right]}{\left(b_2 - b_1\right)\left[\chi^2(b_2) - \chi^2(b_3)\right] - \left(b_2 - b_3\right)\left[\chi^2(b_2) - \chi^2(b_1)\right]}$$

You can then take three values of *b* closer to this new b_{min} and repeat the process. A code is available in *Numerical Recipes* to optimize your choices of b_i and with a couple of extra steps to make sure you don't converge to a local maximum instead of minimum.