Chapter 8: LINEAR REGRESSION

> " 'It is a point of great delicacy, and you must assist us in our endeavours to
> choose exactly the right line of conduct.' "
> --- *Mansfield Park*

8.1 INTRODUCTION

Many physical systems have the happy property that the effect you're investigating has one primary cause and a nice, simple dependence on that cause. The simplest interesting dependence is a linear one; that is, one in which the cause (described by x) and the effect (described by y) are related in this way:

$$y = mx + b$$

This is known as a linear relationship because (as we're actually pretty sure you already know) if you were to graph y against x on a set of coordinate axes, the resulting graph would be a straight line.

Imagine that you're going on a longish road trip (by car), during which you expect to be doing a lot of highway driving at an essentially constant speed. In this case, if you were to measure the odometer reading as a function of the amount of time you've been on the road, you would find that a graph of your results was a straight line. (If the odometer reading *weren't* a linear function of the time, you could quite rightly conclude that you hadn't been driving at a constant speed.) In this example, we can rewrite that general linear equation to fit our specific physical situation *and* give physical interpretations of all the symbols. We might choose letters to remind us of exactly what each symbol represents and write our equation as

$$d = vt + d_0$$

Here t refers to the time on the road (at constant speed), d refers to the distance traveled in that time, and v is the speed (the magnitude of the velocity) of the car. What about that d_0 term? That's the initial reading on the odometer. If you have a trip odometer on your car that you reset to zero at the beginning of a trip, d_0 on the trip odometer would be zero. But unless you buy a new car every time you set out on a road trip, d_0 will almost never be zero if you're using your main odometer for this exercise.

It's possible to work the speedometer/odometer problem in reverse. Suppose your speedometer isn't working properly in that the number the needle is pointing to (OK, most of the physics faculty drive old clunkers with analog speedometers) is not really the speed of the car. It's working well enough that if you keep the needle pinned at 60 miles/hour, your car is traveling at some constant speed; you just can't be confident that the constant speed is in fact 60 mph. This could be pretty important if you're driving through an area dense with highway patrolfolk with quotas of speeding tickets to fill. You could still determine your speed, and calibrate your speedometer at the same time, by holding your speed constant (as registered on your uncalibrated speedometer) and recording the value that the odometer registers at several different times. If both your odometer and your clock were ideal measuring devices, able to register displacement and time without experimental uncertainty, a graph of the odometer values versus time would lie along a perfectly straight line. The slope of this line would be the true speed corresponding to your chosen constant speedometer setting.

In real life, of course, you don't have perfect equipment, and thus your measurements will always include some experimental uncertainty. Therefore, though a graph of real measurement data (such as the data you might get with a *real* car odometer and clock and an unsteady foot on the gas pedal) might still give you a pretty straight line, but your data points would probably *not* all lie exactly on the line. In fact, you might well be able to draw more than one line that looks like a good match to your experimental results.

The purpose of this chapter is to describe a procedure for finding the slope and intercept of the straight line that "best" represents your data in the presence of the inevitable experimental uncertainty of such measurements. Statisticians call the process of determining such a "best fit" line **linear regression**.

8.2 FINDING THE "BEST-FIT" LINE

Before we can determine the line that "best fits" a set of experimental data points, we must first agree on a quantitative criterion for deciding what a "best" fit would be. Several different kinds of criteria might be used, but the most common criterion uses the **least squares** best fit. This criterion says that the "best-fit" line will be the one that *minimizes* the sum of the squared vertical distance between each point and the line. This means that (in some sense at least) that the line is as close as possible to each data point.

To be more specific, suppose that the "true" equation describing your data has the form

$$y = mx + b \tag{8.1}$$

The least squares best fit line is the line with slope m and intercept b that minimizes the sum

$$S \equiv \sum_{i=1}^{N} \left(y_{i,obs} - y_{i,calc}\right)^2 = \sum_{i=1}^{N} \left(y_{i,obs} - mx_i - b\right)^2 \tag{8.2}$$

Here x_i is the value of the independent variable for the ith data point, y_i is the *actually measured* value of the dependent variable for the ith data point, N is the number of data points, and $y_{i,calc}$ is the value for y that you would *calculate* from x_i using a given pair of values for m and b.

If we were to simply pick some values of m and b out of a hat, then we could calculate $y_{i,calc}$ from these values for each value of x_i, subtract it from the actually measured value y_i, square that difference, and then sum over all the data points to find S. We could then pick another pair of values for m and b, compute S for that pair, and so on. After a while, we would find that certain choices of *m* and *b* make *S* smaller, while others make *S* larger. The “best fit” values of *m* and *b* would be those values that make *S* the smallest.

The trial-and-error method for determining *m* and *b* just described would be very tedious to carry out in practice. Happily, application of some multivariable calculus makes this all unnecessary. With some work, one can show that the values of *m* and *b* that yield the smallest possible values of *S* are:

$$m = \frac{N\left(\sum x_i y_i\right) - \left(\sum x_i\right)\left(\sum y_i\right)}{N\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}, \; b = \frac{\left(\sum x_i^2\right)\left(\sum y_i\right) - \left(\sum x_i\right)\left(\sum x_i y_i\right)}{N\left(\sum x_i^2\right) - \left(\sum x_i\right)^2} \tag{8.3}$$

where the sum should be performed from $i = 1$ to N (that is, over all data points). The derivation of these formulas is somewhat involved and employs calculus (at the level of Math 32) that is somewhat beyond the level of this class, so it is available in an optional section (8.8) that you can look at if you are interested. The point to understand here is that we *can* use calculus to find the values of *m* and *b* that minimize the value of *S* without doing a trial-and-error search.

8.3 THE UNCERTAINTY OF THE SLOPE AND INTERCEPT

If each measured value y_i were exactly equal to its “true” value, then equation 8.3 would yield THE best fit line. But if the measured values y_i (not to mention the measured values of x_i) are uncertain, then it is not clear what y_i *ought* to be for a given x_i and therefore what the best-fit line really is: uncertainties in y_i and x_i therefore make *m* and *b* uncertain, too. Now in many cases, *m* and/or *b* are interesting physical quantities whose values we would like to determine in this experiment: if this is so, we would like to know the uncertainty of each of these quantities. How might we determine this uncertainty?

One way to compute the uncertainty would be to repeat the whole experiment, say, 20 times (either sequentially or with 20 different lab teams). If we did this, each data set that we collect would be somewhat different (due to experimental uncertainties), and therefore the best-fit values for *m* and *b* would be somewhat different as well. We could then estimate the uncertainty of *m* in a very straightforward way by taking our 20 different values of *m* (one for each trial of the experiment), finding the standard deviation *s* of these values, and then computing the uncertainty by multiplying *s* by the Student *t*-factor for $N = 20$. We could then calculate the uncertainty of *b* in the same way. This method would yield honest *experimental* estimates of the uncertainties in *m* and *b* that are based directly on what the uncertainties of these quantities *mean* at the fundamental level.

The problem is that actually doing this would be a *very* tedious and time-consuming way to answer the question. Fortunately, we have a shortcut. While we may not have time or person-power to actually do the experiment 20 times, a computer program can easily *simulate* the process of running the experiment 20 times (and thus compute the uncertainties in *m* and *b* as described above) using data from a single real experiment.

8.4 USING THE LINREG APPLICATION

LinReg is a computer program that does this simulation for you. It accepts your experimental data from one run of the experiment you performed, and calculates the best-fit slope *m* and intercept *b* for your actual data. But with the push of a button, you can make the program simulate the efforts of 19 other imaginary lab teams, who gather data with uncertainty ranges matching yours but with actual measured values varying randomly within the uncertainty range specified for each quantity. By computing *m* and *b* for each of the 20 slightly data sets (your actual data and the data from the 19 imaginary teams), the program can easily generate estimates of the uncertainties of *m* and *b* that follow from the uncertainties of your measurements.

LinReg is available for both the Macintosh and Windows operating systems, so to use the program, select a lab computer with your favored operating system and launch the program in the way that is appropriate for that operating system. You can also download either flavor of program from the Physics 51 web site. When the program starts, it displays a window with cells on the left-hand side where you will enter information about your data set.

Enter your names(s), the horizontal variable name and unit, and the vertical variable name and unit, and press the “Done” button. The program will not let you proceed until you entered something in each text box. (If one of your variables is unitless, type “none” in the units box.) After you have successfully done this, the program displays a table in the previously empty part of the window into which you can enter your data. When you type a number, it is entered into the active cell in the table (the white cell). You can change the active cell in the table by pressing the tab key (to scan forward across the table), shift-tab (to scan backwards), return (to move down one row), or the arrow keys.

When you have entered all your data, you can display a graph of your data by pressing the “Graph” button in the “Display Modes” box. If you also press the “Do Linear Fit” button, the program computes and draws the best fit line to your data. The program also uses your table of data as the basis for creating 19 new tables of data that are the simulated experimental results of 19 lab teams like your own. In creating the invented data tables, *LinReg* simulates the measurement process by randomly choosing each measurement value from a bell-shaped distribution of values having the same mean and uncertainty as the corresponding measurement in your table. This creates sets of data that are essentially like yours except that each measurement value has been perturbed from your corresponding value by an amount consistent with your uncertainty, exactly as if a different lab team had made the measurement and come up with a bit different value.

By pressing the up or down arrow buttons on the screen, you can scan through the data from the 19 other imaginary lab teams one team at a time. *LinReg* displays the data for the selected team either as a table (if you are in the table display mode) or a graph (if you are in the graph display mode), just as it would display your data. The number of the team appears at the top of the table or the top left of the graph. Pressing the “Yours” button takes you immediately back to your own (actual) data set.

Also, after you have pressed the “Do Linear Fit” button, *LinReg* will automatically display the values of the slope and intercept for the best-fit line to the displayed data as well as the uncertainties in these quantities. It computes the uncertainties of these quantities in the straightforward manner described in the previous section. To compute the uncertainty in the slope, for example, it simply computes the standard deviation of the 20 best-fit slopes generated from the 20 lab team data sets (your set and the 19 simulated sets) and multiplies by the Student-*t* factor for $N = 20$. (You can easily verify this by checking the calculation by hand if you like.) This is exactly the way that we would calculate the uncertainty of any given team’s slope if we had access to the data from 20 actual lab teams. *LinReg* computes the uncertainty in the intercept the same way.

You cannot change any of the invented data sets, but you can change your own data at any time by displaying your own data using the table display mode, selecting the cell to be changed and retyping the data. If you do this, however, you will find that you have to press the “Do Linear Fit” button again to display the new best-fit line, because *LinReg* now has to recompute the best fit line to fit your modified data. *LinReg* also will recompute all of the invented data sets from the imaginary lab teams to be consistent with your new data.

You can also clear all your data and start over by pressing the “Clear Data” button. You can also at any time change the variable labels and/or units and/or the names of your team members by pressing the “Edit Names and Units” button.

You can also print any table or graph shown on the screen by selecting “Print…” from the File menu. *LinReg* will change the size of the graph to fill a full sheet of paper, but otherwise, what you see on the screen will appear on your printout. The lab computers will be set up to print automatically to the printer in the lab room. If you use *LinReg* on your personal computer, it should automatically print to whatever print destination you have set up for your computer.

The current version of *LinReg* (as of this writing) cannot save your measurement data or read data from a disk file. This was a deliberate design decision (so that the lab computer’s hard disks did not get filled with people’s old *LinReg* data files), but it does mean that if you clear your data or quit *LinReg* by accident, you will have to enter all of your data again by hand.

In addition to being found on the computers in the lab, *LinReg* can be downloaded from the Physics 51 web site (www.physics.pomona.edu/phys51.html). *LinReg* is freeware: you may make a copy for yourself and/or distribute it to your friends.

8.5 USING LINREG TO CREATE LOG-LOG PLOTS

If you check the “Show Ln” check box for either the horizontal or vertical variable, *LinReg* displays and/or plots the natural logarithm (and automatically computes and displays the uncertainty in that logarithm) of each measured value of that variable. This feature allows you to create log-log plots (see Chapter 10) or semilog plots (see Chapter 11) very easily: to create a log-log plot of your data, for example, simply check both boxes!

You will find, however, that you cannot edit the value of the logarithm or its uncertainty. To make changes to your data, uncheck the “Show Ln” box and then modify the data that you originally entered. This is because *LinReg* uses your originally-entered values as the master data set: the logarithms are treated as just a different way to display this master data.

You will also see that *LinReg* computes *natural* logarithms instead of base-ten logarithms. We will use base-ten logs in Chapter 10 because it makes it a bit easier to explain and understand how the logarithms are related to the original values. However, if you think about it, the base one chooses to work in is irrelevant: given a power-law relationship $y = kx^n$, we have

$$\log y = \log k + n \log x \qquad \text{and also} \qquad \ln y = \ln k + n \ln x \qquad (8.4)$$

where log() is the base-ten logarithm and ln() is the natural logarithm. Therefore if we plot ln *y* versus ln *x*, we still get a straight line with slope *n*. The only thing that is different is that the intercept is now equal to ln *k*, not log *k*.

8.6 OTHER TOOLS FOR DOING LINEAR REGRESSION

Many scientific calculators and a number of commonly available spreadsheet or statistical analysis programs can also do linear regression (that is, compute a best-fit line to entered data). However, such programs rarely calculate the uncertainties in the slope and intercept of this line, and even when they do, they usually do it in a way that makes incorrect assumptions (such as assuming that the uncertainties in the horizontal variable are zero and/or the uncertainties in the vertical variable are all the same). We think that you will find that *LinReg* is easier to use than these alternative tools and also will give you better and more useful results in the context of this lab.

We might mention in passing that many of these calculators and/or programs compute a quantity called “correlation coefficient” (usually given the symbol *R* or r^2) for your data when they compute a best-fit line. *R* is a measure of closely your data fit a straight line, and is thus an (indirect) expression of the uncertainty we should have about how good the line is. (*R* = 1 indicates that your data fit a line perfectly; *R* = 0 indicates that there no correlation between your data and any straight line.) While the correlation coefficient *R* can be very useful in helping one discern whether there is a meaningful linear relationship buried in the noisy data sets common in the social sciences, the data in physics experiments usually fit a straight line so well that *R* is always essentially 1, making it pretty useless as a measure of the quality of the best-fit line. Therefore, if you *do* compute *R* for any data in this lab program, don’t brag about how close to 1 it is!

8.7 LIMITATIONS OF LINEAR REGRESSION

It is important to recognize that the technique of linear regression and the program *LinReg* have their limitations. In particular, using *LinReg* makes linear regression so easy that you are likely to forget an important point: the equations and the *LinReg* program will always try to find the “best” line that fits your data, even when your data do not resemble a line at all! *There is no substitute for actually looking at the graphed data to check that it looks like a reasonably straight line*. The pre-lab exercises at the end of this chapter illustrate this point.

8.8 (OPTIONAL) DERIVING EQUATION 8.3

You do not have to understand where equation 8.3 comes from, but its derivation is not all that complicated if you have had Math 32, and understanding the derivation can help you understand the least-squares method better.

Let us rewrite the sum of the squared differences *S* (see equation 8.2) by working out the square in the sum explicitly, as follows:

$$S = \sum_{i=1}^{N} \left(y_i - mx_i - b\right)^2 = \sum_{i=1}^{N} \left(y_i^2 - 2mx_i y_i + m^2 x_i^2 - 2by_i + 2mbx_i + b^2\right) \qquad (8.5)$$

We can minimize *S* the usual way of setting its derivative to zero. Since *S* depends on two unknown quantities (*m* and *b*) here, we must calculate *partial* derivatives of *S* with respect to *m* and *b*, and set each derivative *separately* equal to zero. Remember that to calculate the partial derivative $\partial S / \partial m$ of *S* with respect to *m*, we calculate the ordinary derivative of *S* while treating *b* as a constant. Similarly, to calculate the partial derivative $\partial S / \partial b$ of *S* with respect to *b*, we treat *m* as a constant. Therefore, we want:

$$0 = \frac{\partial S}{\partial m} = \sum_{i=1}^{N} (-2x_i y_i + 2mx_i^2 + 2bx_i) \;\Rightarrow\; \sum_{i=1}^{N} (bx_i + mx_i^2) = \sum_{i=1}^{N} x_i y_i \qquad (8.6a)$$

$$0 = \frac{\partial S}{\partial b} = \sum_{i=1}^{N} (-2y_i + 2mx_i + 2b) \qquad \Rightarrow\; \sum_{i=1}^{N} (b + mx_i) = \sum_{i=1}^{N} y_i \qquad (8.6b)$$

Though they are unknown, we can factor *m* and *b* out of the summation (since their values are the same for all values of *i*). Equations 8.6a and 8.6b then become (respectively):

$$b\sum_{i=1}^{N} x_i + m\sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} x_i y_i \,, \qquad (8.7a)$$

$$b\sum_{i=1}^{N} 1 + m\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i \Rightarrow Nb + m\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i \qquad (8.7b)$$

since the sum of 1 from *i* = 1 to *N* is simply *N*. Now, let us define $A \equiv \sum x_i$, $B \equiv \sum y_i$, $C \equiv \sum x_i^2$, and $D \equiv \sum x_i y_i$: these are *known* quantities since they can be computed from our experimental data. We can rewrite equations 8.7 in terms of these quantities as follows.

$$Ab + Cm = D, \quad Nb + Am = B \qquad (8.8)$$

These represent two equations that we can solve the two unknowns *m* and *b* in the usual way. (For example, to compute *m*, multiply the top equation by *N* and the bottom by *A*, add the equations to eliminate *b*, and then solve for *m*.) The results are:

$$m = \frac{ND - AB}{NC - A^2} = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{N\sum x_i^2 - \left(\sum x_i\right)^2} \qquad (8.9a)$$

$$b = \frac{CB - AD}{NC - A^2} = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{N\sum x_i^2 - \left(\sum x_i\right)^2} \qquad (8.9b)$$

which are equivalent to equations 8.3.

EXERCISES

In the following exercises, plot the data given in the box on the left in the graph area provided to the right. The slope and intercept for the "best" line fitting each data set (according to the least-squares method) appear below the box for the data. Sketch this "best-fit" line on the graph. Then comment: does this line really appear to fit the data? (You will see how important drawing a graph is for successfully interpreting how much real meaning these slope and intercept values have!)

Exercise 8.1

x	*y*	*x*	*y*
0.0	–0.10	6.0	2.69
1.0	0.31	7.0	3.52
2.0	1.05	8.0	3.86
3.0	1.55	9.0	4.51
4.0	1.80	10.0	5.02
5.0	2.41		

$m = 0.51, \quad b = –0.11$

Exercise 8.2

x	*y*	*x*	*y*
0.0	1.22	6.0	6.38
1.0	3.25	7.0	6.13
2.0	4.97	8.0	7.28
3.0	3.68	9.0	6.93
4.0	5.44	10.0	5.30
5.0	4.48		

$m = 0.44, \quad b = –2.83$

Exercise 8.3

x	y	x	y
–10	0	0	6.00
–8	–3.60	2	–5.88
–6	4.80	4	5.50
–4	–5.50	6	–4.80
–2	5.88	8	3.60
0	–6.00	10	0

$m = 0.0$, $b = 0$