LINEAR REGRESSION

Many engineering programs require their students to complete a full semester course on calculus based statistics. Many other programs do not require a statistics course, but expect students to pick up bits and pieces of statistics over several classes. New Mexico Tech falls into the latter of these two scenarios. A student will not be an expert at statistics after only a few of these courses that cover some statistics. However, by the end of the curriculum, they should be competent in statistics such that they can pass the FE Exam.

Linear regression is a topic usually well-covered in statistics courses that is very important to any engineer. Linear regression is not a difficult task to carry out, but to understand and derive the equations used can be challenging. The student in computer programming is expected to be capable of using the equations and hopefully will gain some understanding of the concepts used to derive the resulting equations.

Linear regression is used often by engineers in two different scenarios. A good portion of engineering labs, and science labs for that matter, is to carry out an experiment, collect data, and compare data to "theory". The "theory" is some equation that is supposed to describe what is happening during the experiment. Statistics and linear regression are used to describe how well the experiment fits the data and if the fit is good, then the value of some physical constant can be implied. If the fit is poor, which is usually the case to the chagrin of the student, the student will need to explain what went wrong, what assumption in the derivation of the "theory" was invalid, etc. The second and equally common scenario occurs when an engineer is introduced to an older piece of equipment that needs to be calibrated. Calibration of valves, ovens, furnaces, or similar equipment is quite common. Often the equipment is too old to work with newer, high tech process control software, but the equipment still works and does the job sufficiently well that connection to expensive process control hardware and software is not worth the money.

To better understand linear regression, the topic will be covered in the context of calibrating a furnace. Ideally, a furnace used in a laboratory setting will have an input setting that is equal to a temperature and the output of the furnace is that the temperature inside is equal to the set-point. It sounds simple enough, but it is not necessarily true. In the days when process control software or hardware was expensive and bulky, furnaces were simply given a setting of percent on. This percent is typically the percent of the maximum current or voltage supplied to the heating elements inside the furnace. Since the set-point was adjusted by hand, the operator was the feedback loop to control the temperature. He/she would pick a setting, see if the temperature was too high or too low. If it was too high, the setting would be decreased and if it was too low the setting would be increased. The new setting would be put under the same scrutiny until a final temperature close to the desired temperature was achieved. The difficulty in this technique lies in the fact that a furnace has a slow reaction time. Heating is slow, but if you needed a higher temperature increasing the setting would increase the temperature in a repeatable fashion. If you needed to lower the temperature, the furnace could be turned off and it would take forever. The furnace had means of heating, but no means to cool,

so cooling took forever. To avoid these problems of human process control, the furnace would be calibrated before use. The calibration will yield a line that relates the setting and the temperature using an equation. With the equation, the temperature can be calculated if you know the setting or if you have a desired temperature, the setting can be back calculated.

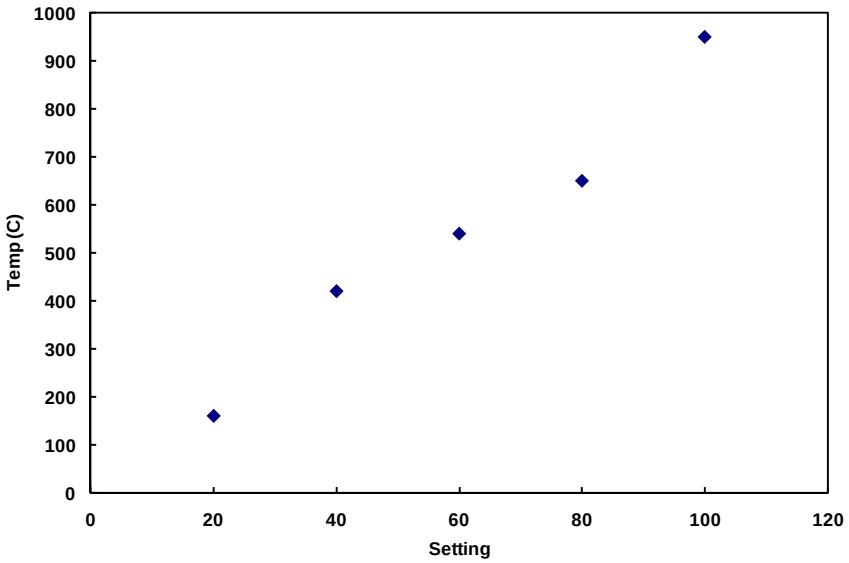
The calibration would be performed by changing the input setting to 10%. Then the temperature of the furnace would increase asymptotically to some final temperature, which corresponds to 10%. It might take a while, but as long as the calibration is carried out only occasionally, it is not a big problem. Then the operator would increase the setting to a new value and the temperature of that setting would be measured. Several data points would be taken. Normally a linear relationship would be assumed between the setting (S) and the output temperature (T) where m is the slope of the line and b is the y-intercept.

$$T = m * S + b$$

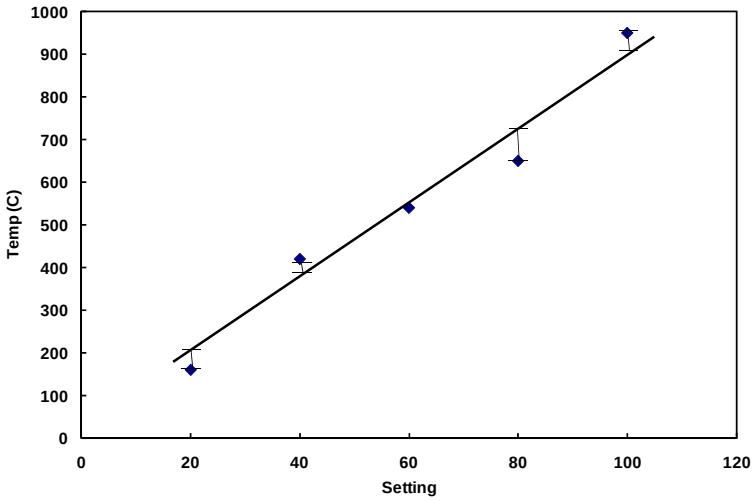
For this case, the y-intercept should be close to the ambient temperature, since it represents the temperature in the furnace when the furnace is off. A set of example data is given below:

Setting (S) [%]	20	40	60	80	100
Temp (T) [°C]	180	380	540	680	880

The data are plotted below. It should be obvious that a straight line, as opposed to a cubic, exponential, or some other line, will describe the relationship quite well.



Now how does one find a calibration line that accurately describes the data? One could simply draw a line through the data and assume that it describes the data accurately.

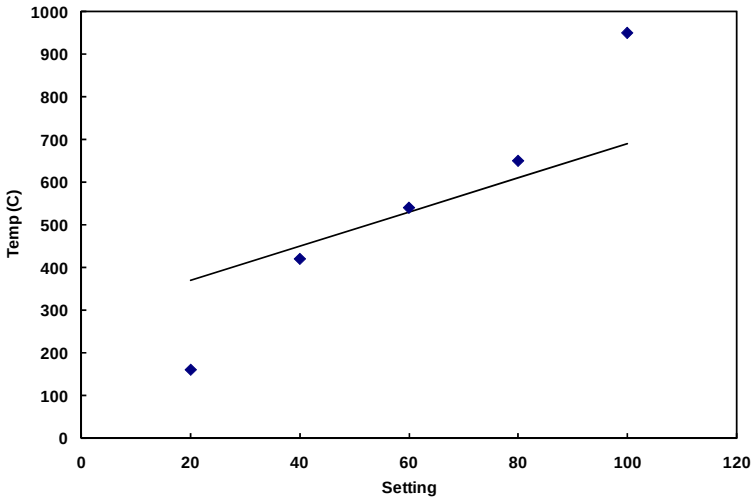


This does not sound mathematical enough for our tastes as engineers, so we need to come up with a way to measure the error of our line. One logical way to measure the error is to take the difference of each data point with the curve fit and add up all of these errors. These errors are shown on the plot above If the error is too large, the slope and intercept could be changed. An equation would look like:

$$error = \sum_i y_i - y_{i,cf}$$

The Greek letter sigma (Σ) sums up the differences between each data point and its corresponding value on the curve fit.

This is not an accepted method. The reason it is not acceptable is that positive and negative errors can cancel out. Take the example below of a randomly drawn curve fit line with a slope of 4 and a y-intercept of 290.



Setting	20	40	60	80	100		SUM
Temp.	160	420	540	650	950		
Bad T Fit	370	450	530	610	690		
Error	-210	-30	10	40	260		70
Error^2	44100	900	100	1600	67600		114300

Clearly the line does not fit the data as well as the other randomly drawn line, but the total error is low. The plot and the data show that the negative errors at the low settings cancel out the positive errors at the higher setting. To fix this the absolute value of the difference can be taken before the summation or the error can be squared. Squaring of the error is preferable to the absolute value for several reasons that are not important to this discussion. The squared error is shown in the table and a large value of the summation is found. It is important to keep in mind that squaring each error and then summing up the squared terms is different than summing all of the terms and squaring that sum. An equation to describe the sum of the squared errors is given below:

$$SSE = \sum_i \left(y_i - y_{i,cf}\right)^2$$

In this equation, *SSE* is the sum of the squared errors, y_i is the data point, and $y_{i,cf}$ is the value on the curve fit that corresponds to a data point. The large greek letter sigma indicates that the error associated with each data point is being added into one sum.

Now, a good understanding of how to evaluate a curve fit has been demonstrated. The question remains, is there a way to find the best fit line for the data? The answer is of course, but it takes knowledge of calculus so the full derivation will be abridged. The student should be able to describe the steps without actually carrying them out. The squared error can be expressed in terms of the independent variable, x_i:

$$SSE = \sum_i \left(y_i - y_{i,cf}\right)^2 = \sum_i (y_i - mx_i - b)^2$$

In this equation, the linear expression for the curve fit has been plugged in for $y_{i,cf}$.

To find the best fit line, the sum of the squared errors should be minimized. In Calculus, students learn that the minimum or maximum of a function can be found by taking the derivative and setting it equal to zero. In other words, the slope of a line at a maximum or minimum is zero. This will be dealt with thoroughly in Calculus, so this description will be left as is. For this problem, the derivative of *SSE* is taken with respect to *m* and a second time with respect to *b*. This yields two equations and two unknowns, *m* and *b*. With proper rearrangement the final result is two equations: one for *m* and one for *b* that depend on sums of the data:

$$m = \frac{n\sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{n\sum_i x_i^2 - (\sum_i x_i)^2}$$

and

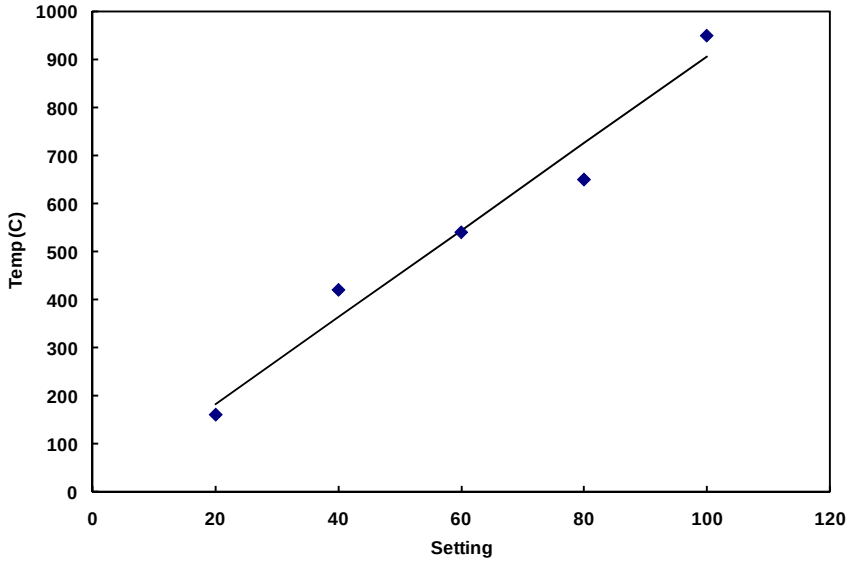
$$b = \frac{\sum_i y_i \sum_i x_i^2 - \sum_i x_i y_i \sum_i x_i}{n\sum_i x_i^2 - (\sum_i x_i)^2}$$

In these equations, n is the number of data points. The summations of x_i^2 and $x_i y_i$ are found by squaring all of the elements of x and then adding them all up and by multiplying x and y element by element then summing them all up.

For our problem of calibrating a furnace, the data table could be expanded to calculate all of these values:

							SUM
S (x)	20	40	60	80	100		300
T (y)	160	420	540	650	950		2720
S*T (x*y)	3200	16800	32400	52000	95000		199400
S^2 (x*x)	400	1600	3600	6400	10000		22000

The slope and y-intercept are found to be 9.05 and 1, respectively. This line can then be plotted over the data to see what the best fit line looks like:



CODING IN MATLAB

Writing an m-file that takes data as an input and returns the slope and intercept of the best fit line is not an extremely difficult task. Most of the calculations required to solve the

problem are summations of the elements of an array. This has been covered in a much earlier section. In fact, the average function written in an earlier tutorial can be modified to output the total and used over and over again in a linear regression m-file. An extremely important thing to remember is that when you call an m-file in another m-file you do not need to use the same input and output names that are used in the original m-file. For example, in class, *T* was the input for our average function, but when the function was called in the command window, any variable name could have been used.

LINEARIZATION

Finally, it is quite common that *x* and *y* data are not linearly related. It is still possible to calculate the sum of the squared errors, but it can be much more difficult. Often, the relationship can be linearized. This most often occurs in the matching theory to experiment scenario engineers see in laboratories. Occasionally, in calibration a linear relationship may not be valid either. To linearize, the expression relating *x* and *y* is manipulated algebraically to yield functions of *x* and *y* that are linearly related. Then these functions of *x* and *y*, not the actual *x* and *y* data, are plugged into the equations for *m* and *b*.

As an example, in chemistry the following relationship is common in lab experiments:

$$y = Ae^{Kx}$$

To linearize this equation the natural logarithm can be taken of both sides to give the following expression, which linearly relates a function of x and a function of y:

$$ln(y) = ln(Ae^{Kx}) = ln(A) + ln(e^{Kx}) = ln(A) + Kx$$

Now, the *ln(y)* will be linearly related to *x*, so in the equations for the slope and intercept the *y* data will not be used, but the *ln(y)* data will be used instead. Also, the slope that is calculated will be equal to *K*, which is some physical constant. The y-intercept is equal to the *ln(A)*, not *A* itself.

This type of relationship is found over and over in first order systems, for which all engineers are exposed to in many different contexts. In general, linearization is easier said than done. It requires trying different algebraic manipulations with the hope that the linear relationship will result. If it does not, then another set of manipulations must be tried.

The coverage of linearization in this tutorial is minimal and not intended to be comprehensive. It is covered so the student has an idea of what to do if the data does not appear to be linear.