Lesson 3: Method of Least Squares & Regression

So far we have looked at how to estimate uncertainty when you have a number of measurements of the same quantity and when you have simple linear graphs. As you can imagine, not every phenomenon in the world can be represented by a linear graph. In this lesson we will look at how computers calculate the trendline equations and how to get uncertainties for these values, which will allows us to gain access to these values for more complicated functions.

Let's start by thinking of a set of data points plotted on an xy-graph. If we were doing this by hand we could estimate by eye what a best fit line would be. We could then try and quantify how well this line fit the data by looking at the vertical distance between the each point and the best fit line. We might then average these distances, but we would run into the same issue as with standard deviation; we do not want the negative distances to cancel out the positive ones. So again, the solution is to use the squared distances.

This technique is know as the method of least squares and it can be used for any plot no matter how strange the curve gets. The computer will try many different possible curves to fit the data and for each one it will look at the sum of the squares of the differences between the curve's y-values and the data points' y-values. The curve that is chosen as the best fit will be the one where this sum is the smallest.

You may ask why we are looking only at the vertical when we could look just as well look at the horizontal or even at a line that is the shortest distance between the point and the curve. This comes down to the idea that the x-variable is independent. As experimenters we have control at which x-values we take our data at, so it is assumed that any error that exists will be in the dependent variable on the y-axis. Now of course this is just an assumption, but in most cases it is a good one.

The procedure that is described above is a very general one and is known as a numerical technique. The computer guesses at a solution, sees how good it is, and tries to make better guesses until it can no longer improve the solution. For simple curves like lines and parabolas there are analytic solutions. So when you plot on Excel, there is a formula that gives the equations presented. However, this exact answer is the same solution you would get if the machine used the numerical method to find it. One advantage to the analytical solution is that every program or computer should give the same solution for the same data set (think Excel vs. a TI calculator).

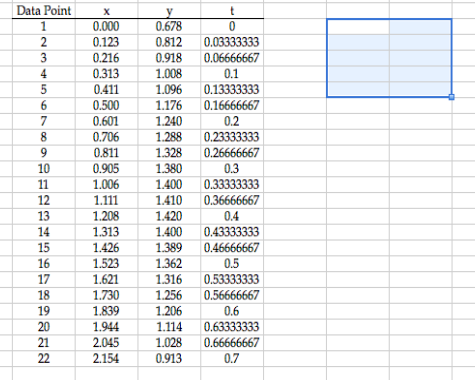
Excel Methods

Even though the graphs that you will be producing this semester have formulas to calculate the equation parameters, we will simply focus on how to use Excel to produce these numbers.

For this exercise you will want to download the file entitled "Projectile Motion Data". We will start to analyze this data during this lesson, but we will return to it several times to make improvements to our estimates.

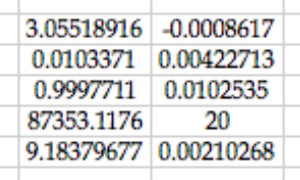
Practice Regression Instructions

- When you open the file you should see data from an experiment similar to lab M2. The *x* and *y* positions of a projectile were recorded every 30th of a second. We will start by examining the graph of *x vs. t*.
- Create a scatter plot of *x vs. t* as you did in lab and add a trendline. You should get *y=3.0552x-0.0009*.
- Now click on an empty cell somewhere on the sheet and highlight a range of cells that is 2 columns wide and five rows high.



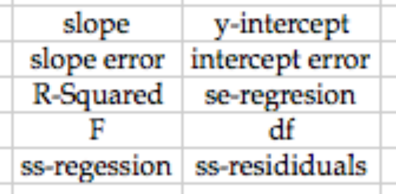
Data Point	x	y	t
1	0.000	0.678	0
2	0.123	0.812	0.03333333
3	0.216	0.918	0.06666667
4	0.313	1.008	0.1
5	0.411	1.096	0.13333333
6	0.500	1.176	0.16666667
7	0.601	1.240	0.2
8	0.706	1.288	0.23333333
9	0.811	1.328	0.26666667
10	0.905	1.380	0.3
11	1.006	1.400	0.33333333
12	1.111	1.410	0.36666667
13	1.208	1.420	0.4
14	1.313	1.400	0.43333333
15	1.426	1.389	0.46666667
16	1.523	1.362	0.5
17	1.621	1.316	0.53333333
18	1.730	1.256	0.56666667
19	1.839	1.206	0.6
20	1.944	1.114	0.63333333
21	2.045	1.028	0.66666667
22	2.154	0.913	0.7

- You then want to use the formula *=linest()*.
  - Type =linest(
  - Highlight the x-position values and type a comma.
  - Highlight the time values and type a comma.
  - Finish the formula by typing "1,1)", **but do not hit enter**. Your formula should look like this…
    =linest(C4:C25,E4:E25,1,1)
  - This function is an array function in Excel, meaning that it will put multiple outputs in different cells. To complete the formula hit the following keystroke.
    - On a PC hit "Shift-CTRL-Enter".
    - On a Mac hit "⌘-Enter".
  - This should give you the following output.



3.05518916	-0.0008617
0.0103371	0.00422713
0.9997711	0.0102535
87353.1176	20
9.18379677	0.00210268

  - Here is the key for this output.



slope	y-intercept
slope error	intercept error
R-Squared	se-regresion
F	df
ss-regession	ss-resididuals

  - Of course we see the slope and y-intercept of our line in the top row. Note that we get the full, unrounded values of these parameters and can click on them to use in other Excel formulas.
  - The second row gives us the standard error for both the slope and the intercept. These errors are equal to an uncertainty at a 68.27% confidence. You would use Table 2 from the previous lesson to get uncertainties at higher confidence.
  - The third row tells us how well the regression model fits the data. The R^2 value is the same as on the graph. This is a value that can range from 0 to 1. An

R^2 of 1 would mean all of the data points fit on the trendline and that *x* completely explains *y*. The *se-regression* is the standard error of the regression. It is the quantification of the general regression technique that we were discussing earlier. If you were to take the average squared distance between the predicted y-values and the actual y-values and then take the square root, you would get this value. The interpretation is that 68.27% of the data points should fall within this amount from the trendline. Again Table 2 can be used to get other intervals. Please note that the standard error of the regression has units of the y-values.

- In the final two rows, the only number we will care about is *df*, which is the number of degrees of freedom. Here we have *N*=22 and we are estimating two parameters: the slope and y-intercept. So in this case *df*=22-2 or 20.

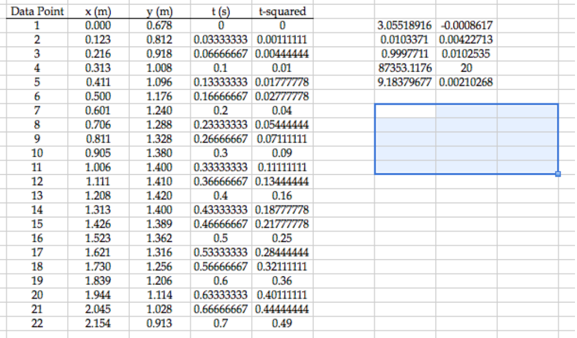
The *linest* function has the following general form.

=linest(known y's, known x's, intercept, stats)

In the example above we wanted to plot the x-position on the y-axis, so we highlighted it first. We then highlighted the times as they were to go on the x-axis. We put in a 1 for both the *intercept* and *stats* argument in the formula. These are binary arguments so they can be 0 or 1. The intercept argument is wanting to know if it should include an intercept or not. As we wanted an intercept we entered 1, if we didn't want one we would have put 0. Likewise, because we wanted the additional statistical information we put a 1 for the *stats* argument.

We can use this same formula to predict some more complicated functions, polynomial for example. Again, plot a graph for comparison. This time we are going to look at *y vs. t*. You should get a formula of *y=-4.9214x²+3.805x+0.6793*.

- Calculate a column next to the time column that is time-squared.
- Then highlight a block of cells three columns wide by five rows high.

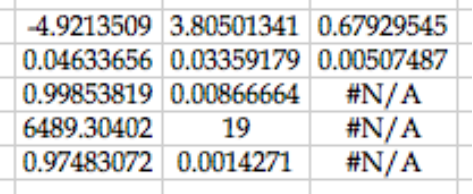


Data Point	x (m)	y (m)	t (s)	t-squared
1	0.000	0.678	0	0
2	0.123	0.812	0.03333333	0.00111111
3	0.216	0.918	0.06666667	0.00444444
4	0.313	1.008	0.1	0.01
5	0.411	1.096	0.13333333	0.01777778
6	0.500	1.176	0.16666667	0.02777778
7	0.601	1.240	0.2	0.04
8	0.706	1.288	0.23333333	0.05444444
9	0.811	1.328	0.26666667	0.07111111
10	0.905	1.380	0.3	0.09
11	1.006	1.400	0.33333333	0.11111111
12	1.111	1.410	0.36666667	0.13444444
13	1.208	1.420	0.4	0.16
14	1.313	1.400	0.43333333	0.18777778
15	1.426	1.389	0.46666667	0.21777778
16	1.523	1.362	0.5	0.25
17	1.621	1.316	0.53333333	0.28444444
18	1.730	1.256	0.56666667	0.32111111
19	1.839	1.206	0.6	0.36
20	1.944	1.114	0.63333333	0.40111111
21	2.045	1.028	0.66666667	0.44444444
22	2.154	0.913	0.7	0.49

3.05518916	-0.0008617
0.0103371	0.00422713
0.9997711	0.0102535
87353.1176	20
9.18379677	0.00210268

- We will use the *linest* formula again. This time highlight the y-positions as the *known y's*.
- Highlight both time and time-squared at once as the *known x's*.
- Put a 1 for both the *intercept* and *stats* arguments.
- You should have the formula =LINEST(D4:D25,E4:F25,1,1)
- Hit the correct keyboard command from above to complete the array formula and

you should get the following output.



-4.9213509	3.80501341	0.67929545
0.04633656	0.03359179	0.00507487
0.99853819	0.00866664	#N/A
6489.30402	19	#N/A
0.97483072	0.0014271	#N/A

- We get all of the same statistics as before but now we have the value of three parameters of the polynomial equation and the associated errors.

We also get three cells that have no information in them. Note that the degrees of freedom has dropped to 19 as we are now estimating three parameters.

Your quiz this week will mostly be more practice of this technique. Can you guess what graph you might be looking at on the quiz?