

Obtaining Uncertainty Measures on Parameters of a Polynomial Least Squares Fit with Excel's LINEST

Faith A. Morrison

Professor of Chemical Engineering
Michigan Technological University, Houghton, MI 39931

24 June 2015

In a previous posting (Morrison, 2014) we described how to use Excel's LINEST to obtain a least squares fit to a linear function, $y = mx + b$. LINEST may also be used to produce fits to polynomials and other nonlinear functions. We describe how to do this in this document.

Fitting to a Straight Line

We give a sketch of the *Normal equation* method here, beginning with the linear case and then adapting the linear case to the polynomial case. Although Excel versions after 2003 do not use the Normal equation method for solving the least squares problem, we feel that discussing this method helps us to understand the structure of the calculation that Excel enables us to do. See the Microsoft support pages (Microsoft, 2015) for more on how Excel is programmed.

The ordinary least squares problem may be posed as a matrix calculation (Co, 2013). Consider the problem of a data set of n data pairs (x_i, y_i) that we would like to fit to the linear function $y = mx + b$. We can construct the following matrices:

$$\text{Function:} \quad y = mx + b \quad (1)$$

$$n \times 2 \text{ matrix:} \quad A \equiv \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \quad (2)$$

$$n \times 1 \text{ matrix:} \quad B \equiv \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (3)$$

$$2 \times 1 \text{ matrix of coefficients:} \quad \alpha \equiv \begin{pmatrix} m \\ b \end{pmatrix} \quad (4)$$

If the data fit the line perfectly, then each data pair would satisfy the equations $y_i = mx_i + b$, and in matrix form we could write:

(all data pairs sit on the line)

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (5)$$

$$A\alpha = B \quad (6)$$

The usual case is that the data do not exactly sit on a line, so the equations $y_i = mx_i + b$ would not hold. Instead, for each data pair there would be a residual amount r_i that is the difference between the line and the data point. This may be written in matrix form as:

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \equiv \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \quad (7)$$

$$A\alpha - B = r$$

Our goal is to find the column matrix α that minimizes the overall magnitude of the set of residuals, r , for our data set (x_i, y_i) .

It can be shown (Co, 2013) that the minimum least-squared summed residual for the data set (x_i, y_i) is given by the following matrix equation, which is called the *Normal equation*:

Normal equation: $A^T A \alpha = A^T B$ (8)

Solving the Normal equation for α gives us the ordinary least squares best fit values of the model coefficients (slope and intercept) for the data set. The Normal equation is a linear equation, which we can solve by left-multiplying both sides by the matrix inverse $(A^T A)^{-1}$:

$$(A^T A)^{-1} (A^T A) \alpha = (A^T A)^{-1} A^T B \quad (9)$$

$$\alpha = (A^T A)^{-1} A^T B \quad (10)$$

The calculation in equation 10 is straightforward to compute using Excel's matrix operations or with other software such as Matlab or Mathematica. LINEST is carrying out this type of calculation.¹

Fitting to a Polynomial

We can easily expand the Normal-equation method to polynomials of higher order. Consider the problem of a data set of n data pairs (x_i, y_i) that we would like to fit to the polynomial function $y = a_2 x^2 + a_1 x + a_0$. We can construct the following matrices:

¹ Excel could use the Normal equation to calculate α and did do so in earlier versions, but to obtain greater stability with some data sets it actually uses a different calculation method. See the Microsoft support pages for details (Microsoft, 2015).

$$\text{Function: } y = a_2x^2 + a_1x + a_0 \quad (11)$$

$$n \times 3 \text{ matrix: } A \equiv \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{pmatrix} \quad (12)$$

$$n \times 1 \text{ matrix: } B \equiv \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (13)$$

$$3 \times 1 \text{ matrix of coefficients: } \alpha \equiv \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} \quad (14)$$

Following the same steps that we used for the fit to the straight line, we construct a column matrix of residuals and minimize the sum of squared residuals (steps not shown; see Co, 2015) by solving the Normal equation for the matrix α .

$$\begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n^2 & x_n & 1 \end{pmatrix} \begin{pmatrix} a_2 \\ a_1 \\ a_0 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \equiv \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} \quad (15)$$

$$A\alpha - B = r \quad (16)$$

$$\text{Result when residuals are minimized: } \alpha = (A^T A)^{-1} A^T B \quad (17)$$

The solution to the Normal equation is written the same for the straight line fit (equation 10) and for the polynomial fit (equation 17). The expansion to a polynomial fit required only the modification of the matrices A and α : In the polynomial fit the matrix A has an extra column on the left containing the x_i values squared (see equation 12). Although here we used a quadratic model (second order polynomial) for the fit, it is straightforward to see that higher order polynomials are easily accommodated by further expanding the A and α matrices in an analogous way. We can also replace the functions in the columns with any other nonlinear functions that interest us, for example $\ln(x)$ (see appendix).

Polynomial Fits with LINEST

The MS Excel function LINEST carries out an ordinary least squares calculation (see Microsoft, 2014; Morrison, 2014). The earlier handout gave instructions for using LINEST to fit data to a straight line; here we discuss fitting a second-order polynomial to a data set. The steps are similar for both linear and quadratic fits; to request the polynomial fit instead of the linear fit we just need to provide the extra column in the A matrix containing the x_i^2 values, as discussed in the previous section.

To calculate the polynomial fit for the data shown in Figure 1, we apply LINEST as follows (instructions are for a PC):

1. Select a blank range of five rows (to accommodate the statistics) by **three** columns (the number of coefficients in the model equation, 3 for quadratic, 4 for cubic, etc.). These cells will store the output of the function; we chose C2:E6 as shown shaded in Figure 1.
2. Click on *Formulas* and then "Insert Function."
3. In the *Insert Function* window, choose category "Statistical" and function "LINEST;" then click on OK.
4. Select the y range; select the x-range (in the example B11:B46) **but create the extra column in the internal LINEST matrix A by appending the following array instructions to your selection for the x-range: x-range^{1,2}**. For the example shown the entry would be B11:B46^{1,2}. **Note the curly brackets; these must be typed in.** The "B11:B46^{1,2}" instruction for the x-range is creating two columns to use in the calculation, one with the x-values to the power "1" and the other a column with the x-values to the power "2". Compare this step with the discussion around equation 12.²
5. For "Const," enter TRUE (TRUE=allow a constant term to be calculated, the a_0 in equation 11) and for "Stats," also choose TRUE (TRUE=show the statistics associated with the fit); click on OK.
6. Specify that LINEST is an array function by selecting the formula *in the entry field* and pressing CTRL-SHIFT-ENTER. The selected output cells will populate with the coefficients of the fit (a_2, a_1, a_0) along with the statistics associated with the fit (see Figure 1).

A good check on our work is to use the *Trendline* command in the Excel plotting windows to fit the data to a polynomial; the fit obtained is the same. We suggest that you do this check and compare the coefficients and R^2 values obtained in the two different methods.

It is possible to directly access the outputs of in a LINEST fit by noting their location in the output array (Table 1). In the example discussed, the results are stored in a 5×3 element array located at C2:E6. We can access each of the 15 outputs individually (including the #N/A indications) with commands in this format placed in any desired cell:

$$=INDEX(LINEST(y\text{-range}, x\text{-range}^{\{1,2\}}, 1, 1), \text{row}, \text{column}) \quad (18)$$

The LINEST call used to create the array in equation 18 is the same as that used earlier in the example, with "1" replacing "TRUE" twice ("1" is equivalent to "TRUE" in logical functions). Note that the command in equation 18 is not an efficient use of computing power since Excel will recalculate the fit for each call of such a command.

² Note that to make the fit to a cubic equation, select a range of 5 rows by 4 columns in step 1 and in step 4 use $x\text{-range}^{\{1,2,3\}}$.

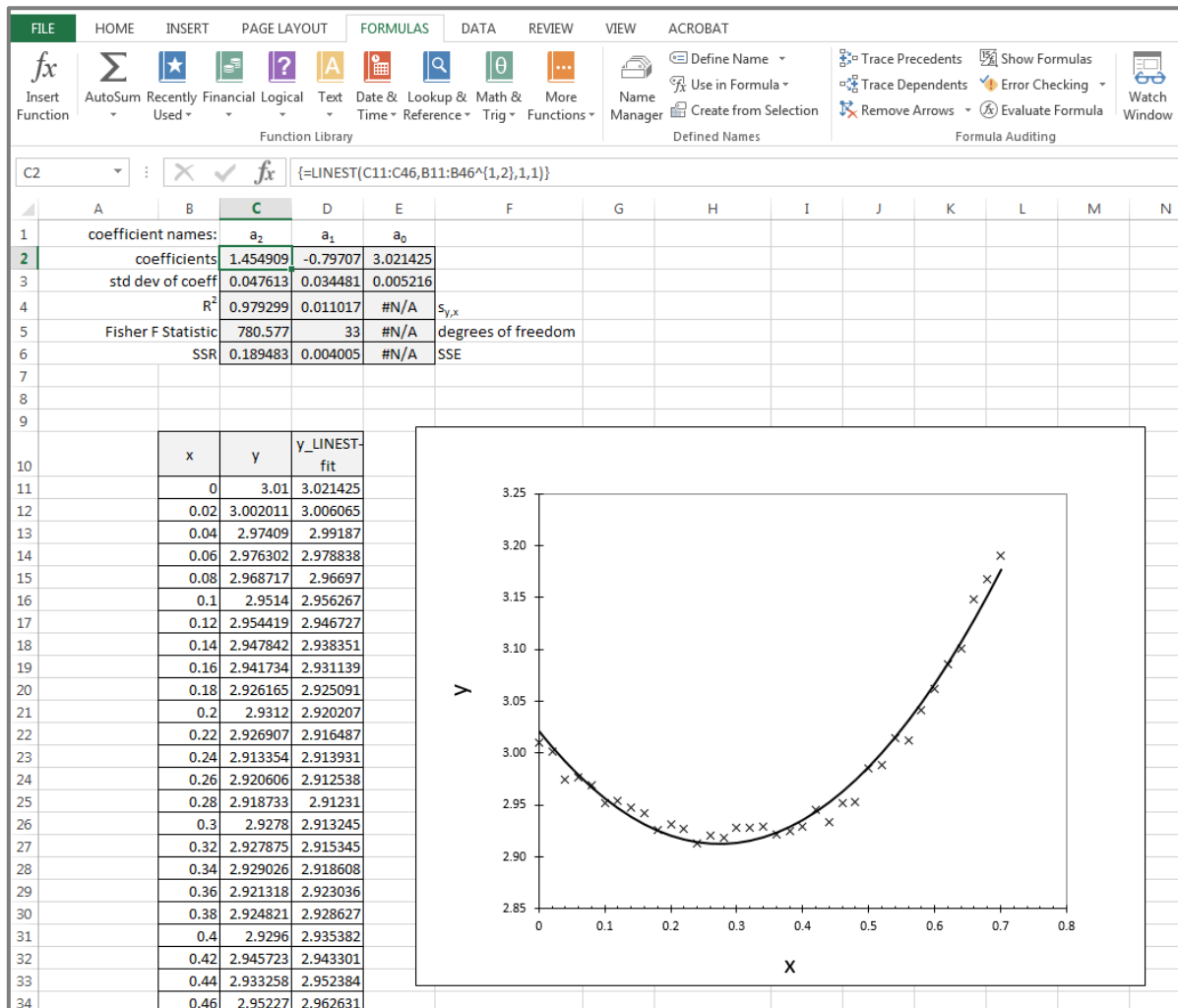


Figure 1: Data (×) were fit to a quadratic equation using Excel’s LINEST function, as discussed in the text. Cells C2 through E6 hold the array output of LINEST. The cells that surround the output cells contain labels that were added manually. Note that in step 6 when we transform LINEST into an *array* function, curly brackets { } appear automatically around the entire function call. These brackets cannot be entered manually, but must be introduced as instructed in step 6. In contrast, the curly brackets in the “ $\wedge\{1,2\}$ ” portion of the command were added manually.

Table 1: Contents of the 5×3 array of output from LINEST for fitting a 2nd order polynomial to a dataset.

Row	Column	Quantity	Description
1	1	a_2	Coefficient of x^2 term
	2	a_1	Coefficient of x term
	3	a_0	Coefficient of constant term
2	1	s_{a_2}	Standard deviation of a_2
	2	s_{a_1}	Standard deviation of a_1
	3	s_{a_0}	Standard deviation of a_0
3	1	R^2	Coefficient of determination
	2	$s_{y,x}$	Standard deviation of $y(x)$
	3	#N/A	-----
4	1	F	Fisher F statistic
	2	d.f.	Degrees of freedom
	3	#N/A	-----
5	1	SSR	Regression sum of squares
	2	SSE	Residual sum of squares of errors
	3	#N/A	-----

References

Tomas B. Co, *Methods of Applied Mathematics for Engineers and Scientists* (Cambridge University Press, New York, 2013).

Microsoft, Inc., "Description of the LINEST function in Excel 2003 and in later versions of Excel," Redmond, Washington, USA, available at <https://support.microsoft.com/en-us/kb/828533>, accessed 22 June 2015.

Faith A. Morrison, "Obtaining Uncertainty Measures on Slope and Intercept of a Least Squares Fit with Excel's LINEST," Department of Chemical Engineering, Michigan Technological University, Houghton, Michigan, USA, 25 September 2014, available at www.chem.mtu.edu/~fmorriso/cm3215/Lectures/CM3215_LectureStatistics_5_LINEST2014.pdf, accessed 22 June 2015.

Appendix

Summary of Some Handy Microsoft Excel LINEST Commands

Linear Fit

$$y = mx + b$$

$$m = \text{SLOPE}(\text{yrange}, \text{xrange})$$

$$b = \text{INTERCEPT}(\text{yrange}, \text{xrange})$$

Logarithmic Fit

$$y = a \ln(x) + b$$

$$a = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{LN}(\text{xrange})), 1, 1)$$

$$b = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{LN}(\text{xrange})), 1, 2)$$

Power-Law Fit

$$y = ax^b$$

$$a = \text{EXP}(\text{INDEX}(\text{LINEST}(\text{LN}(\text{yrange}), \text{LN}(\text{xrange})), 1, 2))$$

$$b = \text{INDEX}(\text{LINEST}(\text{LN}(\text{yrange}), \text{LN}(\text{xrange})), 1, 1)$$

Exponential Fit

$$y = ae^{bx}$$

$$a = \text{EXP}(\text{INDEX}(\text{LINEST}(\text{LN}(\text{yrange}), \text{xrange}), 1, 2))$$

$$b = \text{INDEX}(\text{LINEST}(\text{LN}(\text{yrange}), \text{xrange}), 1, 1)$$

2ND Order Polynomial Fit (Quadratic)

$$y = a_2x^2 + a_1x + a_0$$

$$a_2 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2\}}), 1, 1)$$

$$a_1 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2\}}), 1, 2)$$

$$a_0 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2\}}), 1, 3)$$

3RD Order Polynomial Fit (Cubic)

$$y = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$a_3 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2,3\}}), 1, 1)$$

$$a_2 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2,3\}}), 1, 2)$$

$$a_1 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2,3\}}), 1, 3)$$

$$a_0 = \text{INDEX}(\text{LINEST}(\text{yrange}, \text{xrange}^{\{1,2,3\}}), 1, 4)$$