Best Polynomial Models Using Excel Linest Function

By Namir C. Shammas

In this article I present two versions of VBA subroutines that select the best polynomial regression models among a larger set of models. In both cases, the subroutines obtain the values of observed variables, apply a number of transformations on each variable, and then use these transformed values to determine which combination of transformations yield the best set of fitted polynomials. In other words, the VBA subroutines go beyond performing simple/typical polynomial regression. This approach allows you to investigate polynomial relations between the transformations of the observed variables! While we are certainly interested in the very best polynomial model, it is a very wise to view the best say 20 models. The reason is that if the user has several sets of similar data (each having its own errors) then a possible best empirical polynomial model will consistently appear in the set of top models. Once you select one or more polynomial models, you can use subroutine **MLR**, which I present in the first article, to obtain the regression ANOVA table for these selected polynomial models.

This article presents two approaches for obtaining the best (empirical) polynomial models:

- Scheme P1: Applying a combination of powers and the natural logarithm to the observed variables. These transformed variables then participate in the polynomial regression for a specific polynomial order.
- Scheme P2: Is a version of scheme P1 that covers a range of orders.

The Scheme P1 Approach for Best Polynomial Models

The best way to present and explain scheme P1 is to show you the worksheet for this approach to getting the best polynomial models. Figures 1 and 2 show the left and right sides, respectively, of a sample scheme P1 worksheet. Notice the following individual cells and columns of cells that provide input for the regression program:

- Cell A2 specifies the polynomial order.
- Cell A4 contains the values for the maximum number of best polynomial models. The current value is 20.
- Cells A6 and A8 specify the shift and scale values for the observed independent variable X. The default settings for the shift and scale values are 0 and 1, respectively.
- Cells A10 and A12 specify the shift and scale values for the observed dependent variable Y. The default settings for the shift and scale values are 0 and 1, respectively.
- Column B has the values of the observed dependent variable Y, starting at cells B2.
- Column C stores the values of the observed independent variable X, starting at cell C2.

- Column D MUST BE empty.
- Column E contains the numerically coded transformations for variable Y. The numerical codes for the transformations are the same ones used in the best multiple regression programs.
- Column F contains the numerically coded transformations for variable X.
- Column G MUST BE empty.

The following worksheet columns provide the regression output generated by running the VBA macro **BestPolyReg**:

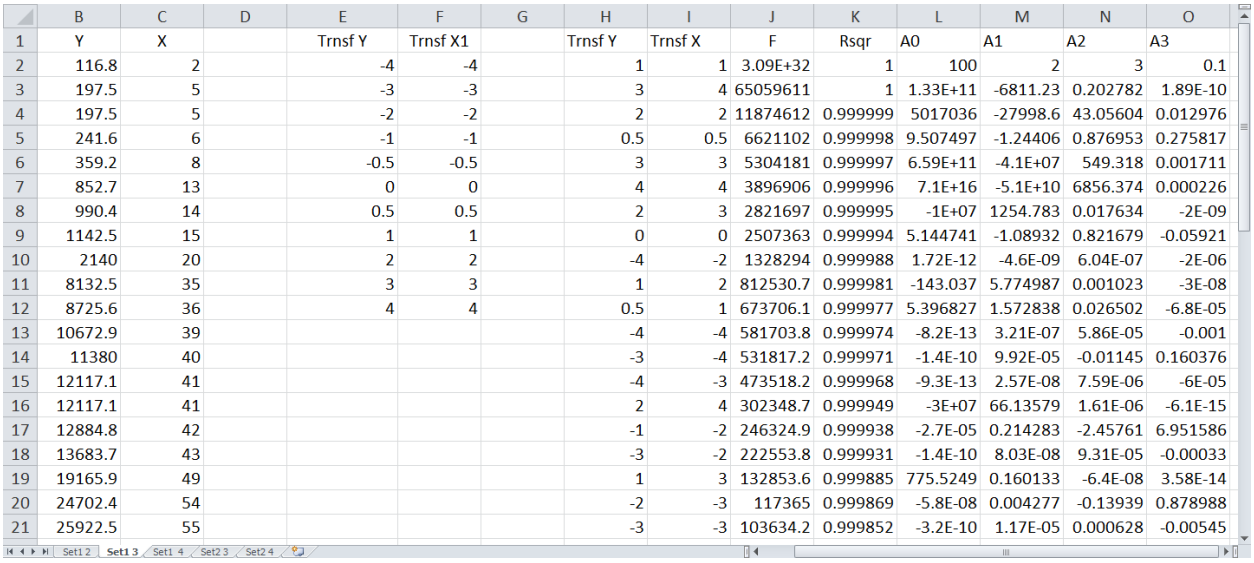
- Column H shows the best transformations for variable Y.
- Column I shows the best transformations for variable X.
- Column J displays the values of the F statistic.
- Column K shows the values of the R^2 statistic.
- Column L displays the values of the intercepts.
- Column M shows the values for regression coefficient for the transformed variable raised to power 1.
- Column N and up displays the values for regression coefficient for the transformed variable raised to power 2 and up, respectively.

The rows that start with the column H represent the record for each model. These rows are sorted in a descending order using the values of the F statistic (in column J).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Polynomial Order	Y	X		Trnsf Y	Trnsf X1		Trnsf Y	Trnsf X	F	Rsqr	A0	A1
2	3	116.8	2		-4	-4		1	1	3.09E+32	1	100	
3	Max Results	197.5	5		-3	-3		3	4	65059611	1	1.33E+11	-6811.2
4	20	197.5	5		-2	-2		2	2	11874612	0.999999	5017036	-27998
5	Shift X	241.6	6		-1	-1		0.5	0.5	6621102	0.999998	9.507497	-1.244(
6	0	359.2	8		-0.5	-0.5		3	3	5304181	0.999997	6.59E+11	-4.1E+(
7	Scale X	852.7	13		0	0		4	4	3896906	0.999996	7.1E+16	-5.1E+1
8	1	990.4	14		0.5	0.5		2	3	2821697	0.999995	-1E+07	1254.78
9	Shift Y	1142.5	15		1	1		0	0	2507363	0.999994	5.144741	-1.089
10	0	2140	20		2	2		-4	-2	1328294	0.999988	1.72E-12	-4.6E-(
11	Scale Y	8132.5	35		3	3		1	2	812530.7	0.999981	-143.037	5.77498
12	1	8725.6	36		4	4		0.5	1	673706.1	0.999977	5.396827	1.57283
13		10672.9	39					-4	-4	581703.8	0.999974	-8.2E-13	3.21E-(
14		11380	40					-3	-4	531817.2	0.999971	-1.4E-10	9.92E-(
15		12117.1	41					-4	-3	473518.2	0.999968	-9.3E-13	2.57E-(
16		12117.1	41					2	4	302348.7	0.999949	-3E+07	66.135
17		12884.8	42					-1	-2	246324.9	0.999938	-2.7E-05	0.2142
18		13683.7	43					-3	-2	222553.8	0.999931	-1.4E-10	8.03E-(
19		19165.9	49					1	3	132853.6	0.999885	775.5249	0.16013
20		24702.4	54					-2	-3	117365	0.999869	-5.8E-08	0.00427
21		25922.5	55					-3	-3	103634.2	0.999852	-3.2E-10	1.17E-(

Set1 2 | Set1 3 | Set1 4 | Set2 3 | Set2 4

Figure 1. The left side of worksheet for scheme P1.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Y	X		Trnsf Y	Trnsf X1		Trnsf Y	Trnsf X	F	Rsqr	A0	A1	A2	A3
2	116.8	2		-4	-4		1	1	3.09E+32	1	100	2	3	0.1
3	197.5	5		-3	-3		3	4	65059611	1	1.33E+11	-6811.23	0.202782	1.89E-10
4	197.5	5		-2	-2		2	2	11874612	0.999999	5017036	-27998.6	43.05604	0.012976
5	241.6	6		-1	-1		0.5	0.5	6621102	0.999998	9.507497	-1.24406	0.876953	0.275817
6	359.2	8		-0.5	-0.5		3	3	5304181	0.999997	6.59E+11	-4.1E+07	549.318	0.001711
7	852.7	13		0	0		4	4	3896906	0.999996	7.1E+16	-5.1E+10	6856.374	0.000226
8	990.4	14		0.5	0.5		2	3	2821697	0.999995	-1E+07	1254.783	0.017634	-2E-09
9	1142.5	15		1	1		0	0	2507363	0.999994	5.144741	-1.08932	0.821679	-0.05921
10	2140	20		2	2		-4	-2	1328294	0.999988	1.72E-12	-4.6E-09	6.04E-07	-2E-06
11	8132.5	35		3	3		1	2	812530.7	0.999981	-143.037	5.774987	0.001023	-3E-08
12	8725.6	36		4	4		0.5	1	673706.1	0.999977	5.396827	1.572838	0.026502	-6.8E-05
13	10672.9	39					-4	-4	581703.8	0.999974	-8.2E-13	3.21E-07	5.86E-05	-0.001
14	11380	40					-3	-4	531817.2	0.999971	-1.4E-10	9.92E-05	-0.01145	0.160376
15	12117.1	41					-4	-3	473518.2	0.999968	-9.3E-13	2.57E-08	7.59E-06	-6E-05
16	12117.1	41					2	4	302348.7	0.999949	-3E+07	66.13579	1.61E-06	-6.1E-15
17	12884.8	42					-1	-2	246324.9	0.999938	-2.7E-05	0.214283	-2.45761	6.951586
18	13683.7	43					-3	-2	222553.8	0.999931	-1.4E-10	8.03E-08	9.31E-05	-0.00033
19	19165.9	49					1	3	132853.6	0.999885	775.5249	0.160133	-6.4E-08	3.58E-14
20	24702.4	54					-2	-3	117365	0.999869	-5.8E-08	0.004277	-0.13939	0.878988
21	25922.5	55					-3	-3	103634.2	0.999852	-3.2E-10	1.17E-05	0.000628	-0.00545

Figure 2. The right side of worksheet for scheme P1.

Looking at columns E and F in Figure 1, you see the list of transformations for each variable. The numerically coded transformations represent powers used to raise the variables in order to calculate their transformations. For example, the value of –4 in cell E2 tells the program that one of the transformations for variable Y is 1/Y^4. You will no doubt notice that columns E and F contain the value of 0. The VBA code treats 0 as a special case and applies the natural logarithm to the corresponding variable.

The transformations of the observed variables require that you obey the following simple rules:

- Each variable has its own set of transformations that are independent, in number and sequence, of those used with the other variable.
- The transformation can be integers and non-integers, as well as negative, zero, or positive. The VBA subroutine has an error handler and will catch runtime errors. Any set of transformation that generates a runtime error is out of the contest, so to speak!

Looking at the sample results in Figure 2, the best model is:

$$Y = 100 + 2\,X + 3\,X^2 + 0.1\,X^3$$

This model has the F statistic of 3.09E+32 and the R^2 value of 1.

The BestPolyReg VBA Subroutine

Listing 1 shows the source code for subroutine **BestPolyReg**. This subroutine performs the following general tasks:

- Read the input from the leading columns. This includes the observations for the different variables, the scale and shift values for X and Y, and the lists of numerically coded transformations.

- Start nested loops to transform the X and Y variables. The nested loops perform the nest tasks.
- Prepare the regression variables for the polynomial regression.
- Invoke the **Linest** function and store its results in variable **vRegResultsMat**. The subroutine compares the value of the F statistic with the lowest value of F in the Hit Parade list. If the newly calculated F statistic is greater than that lowest F value, the subroutine overwrites the last Hit Parade row with newly calculated data and then invokes Excel's Sort method to quickly re-sort the Hit Parade list using values of the F statistics.

```
Option Explicit
Option Base 1

Sub BestPolyReg()

  Const MAX_ERRORS As Double = 1000000# ' initial max error value

  Dim ErrorCounter As Double, MaxErrors As Double, sMaxErr As String
  Dim NumIndpVars As Integer ' number of variables
  Dim PolyOrder As Integer ' polynomial order
  Dim TN As Integer ' total number coefficients
  Dim N As Integer ' number of data points
  Dim MaxTrans As Integer ' max transformations
  Dim MaxResults As Integer ' max results
  Dim Col1 As Integer, Col2 As Integer, Col3 As Integer
  Dim Col4 As Integer, Col5 As Integer
  Dim I As Integer, J As Integer, K As Integer
  Dim M1 As Integer, M As Integer
  Dim iY As Integer, iX As Integer
  Dim TransfMat() As Double
  Dim CurrentTransf() As Double, CountTransf() As Integer
  Dim NumTransf() As Integer ' number of transformations
  Dim Y() As Variant, X() As Variant
  Dim Yt() As Variant, Xt() As Variant
  Dim Xtp() As Variant
  Dim vRegResultsMat As Variant
  Dim F As Double, Rsqr As Double, xval As Double
  Dim ShiftX As Double, ShiftY As Double
  Dim ScaleX As Double, ScaleY As Double

  ErrorCounter = 0
  MaxErrors = MAX_ERRORS
  NumIndpVars = 2
  PolyOrder = [A2].Value
```



```
  MaxResults = [A4].Value
  TN = PolyOrder + 1
  ShiftX = [A6].Value
  ScaleX = [A8].Value
  ShiftY = [A10].Value
  ScaleY = [A12].Value
  Col1 = 2              ' first column of data
  Col2 = 5  ' first column of transformations
  Col3 = 8  ' first column of best transformations
  Col4 = 10 ' first column of results
  Col5 = Col4 + TN + 1  ' last column of results

  Range(Cells(1 + Col3), Cells(1, 50)).Value = ""
  Cells(1, Col3) = "Transf Y"
  Cells(1, Col3 + 1) = "Transf X"
  Cells(1, Col4) = "F"
  Cells(1, Col4 + 1) = "Rsqr"
  For I = 0 To PolyOrder
   Cells(1, Col4 + 2 + I) = "A" & I
  Next I

  Range(Cells(2, Col3), Cells(1 + 2 * MaxResults, Col4 + 3 *
TN)).Value = ""
  Range(Cells(2, Col4), Cells(1 + MaxResults, Col4 + 1 + TN)).Value =
0
  MaxTrans = Range(Cells(2, Col2), Cells(1,
Col2)).CurrentRegion.Rows.Count - 1
  ReDim NumTransf(NumIndpVars), TransfMat(NumIndpVars, MaxTrans),
CurrentTransf(NumIndpVars)
  ReDim CountTransf(NumIndpVars)

  For I = 1 To NumIndpVars
    J = 2
    Do While Trim(Cells(J, Col2 + I - 1)) <> ""
      TransfMat(I, J - 1) = Cells(J, Col2 + I - 1)
      J = J + 1
    Loop
    NumTransf(I) = J - 2
  Next I

  N = Range("B1").CurrentRegion.Rows.Count - 1
  Y = Range("B2:B" & N + 1).Value
  X = Range(Cells(2, 3), Cells(N + 1, 2 + PolyOrder)).Value
  Yt = Range("B2:B" & N + 1).Value
  Xt = Range(Cells(2, 3), Cells(N + 1, 2 + PolyOrder)).Value
```



```
  ReDim Xtp(N, PolyOrder) ' polynomial data matrix

  For iY = 1 To NumTransf(1)
    CurrentTransf(1) = TransfMat(1, iY)
    For iX = 1 To NumTransf(2)
      CurrentTransf(2) = TransfMat(2, iX)

      On Error GoTo HandleErr

      For I = 1 To N
        DoEvents

        If CurrentTransf(1) <> 0 Then
          Yt(I, 1) = (ScaleY * Y(I, 1) + ShiftY) ^ CurrentTransf(1)
        Else
          Yt(I, 1) = Log(ScaleY * Y(I, 1) + ShiftY)
        End If

        If CurrentTransf(2) <> 0 Then
          Xt(I, 1) = (ScaleX * X(I, 1) + ShiftX) ^ CurrentTransf(2)
        Else
          Xt(I, 1) = Log(ScaleX * X(I, 1) + ShiftX)
        End If

        For J = 1 To PolyOrder
          Xtp(I, J) = Xt(I, 1) ^ J
        Next J

      Next I

      ' perform the regression calculations
      vRegResultsMat = Application.WorksheetFunction.LinEst(Yt, Xtp,
True, True)

      Rsqr = vRegResultsMat(3, 1)
      F = vRegResultsMat(4, 1)
      ' check if F > F of last result
      If F > Cells(MaxResults + 1, Col4) Then
        M1 = MaxResults + 1
        ' write new results to row M
        Cells(M1, Col4) = F
        Cells(M1, Col4 + 1) = Rsqr
        For I = 1 To TN
          Cells(M1, Col4 + I + 1) = vRegResultsMat(1, TN - I + 1)
        Next I
```



```
        For I = 1 To NumIndpVars
          Cells(M1, Col3 + I - 1) = CurrentTransf(I)
        Next I

        Range(Cells(2, Col3), Cells(MaxResults + 1, Col5)).Select
        Range(Cells(2, Col3), Cells(MaxResults + 1, Col5)).Sort
Key1:=Range(Cells(2, Col4), Cells(MaxResults + 1, Col4)),
Order1:=xlDescending

      End If ' If F > Cells(MaxResults + 1, Col3)

    GoTo Here

HandleErr:
  ErrorCounter = ErrorCounter + 1
  If ErrorCounter > MaxErrors Then
    If MsgBox("Reached maximum error limits of " & ErrorCounter &
vbCrLf & _
    "Want to stop the process?", vbYesNo + vbQuestion, "Confirmation
requested") = vbYes Then
      Exit Sub
    Else
      sMaxErr = InputBox("Update maximum number of errors", "Max
Errors Input", MaxErrors)
      If Trim(sMaxErr) = "" Then
        MsgBox "User canceled calculations process", vbOKOnly +
vbInformation, "End of Process"
        Exit Sub
      End If
      MaxErrors = CDbl(sMaxErr)
      ErrorCounter = 0
    End If
  End If
  Resume Here
Here:
    Next iX
  Next iY


  MsgBox "Done", vbOKOnly + vbInformation, "Success!"
End Sub
```

Listing 1. The BestPolyReg subroutine.

Handling Runtime Error in Scheme P1, and P2

The subroutines **BestPolyReg** and **BestPolRegSet** (which I will introduce later in this article) have an error handler and an error counting scheme. The constant **MAX_ERRORS** represents the initial value for the maximum number of occurring runtime errors. The subroutine assigns the value of this constant to the variable **MaxErrs**. The subroutine also uses the variable **ErrorCounter** to count the number of occurring runtime errors. When the value in variable **ErrorCounter** exceeds that in variable **MaxErrs**, the subroutine displays a message box telling you that the number of occurring runtime errors has reached its maximum limit. This message should not convey a mandatory end to the calculations. The message box has a **Yes** and **No** buttons allowing you to end the program by clicking the **Yes** button. If you click the **No** button, the subroutine displays an input box prompting you to enter a new maximum error limit. The input box shows the current maximum error limit as the default value. You can alter the maximum error limit or simply press the **OK** button to accept it and proceed with the statistical calculations. In this case, the subroutine resets the value in **ErrorCounter** to 0. You can also stop the subroutine altogether by clicking the **Cancel** button. Thus, the subroutine offers you two points of exit.

The Scheme P2 Approach for Best Polynomial Models

Scheme P2 allows you to find the best polynomial order (in a range of orders that you specify) and the best power/logarithm transformations for each polynomial. Unlike, scheme P1, scheme P2 works on multiple worksheets. Therefore you will need a separate workbook for each set of data when using scheme P2. The first sheet in the workbook specifies the polynomial order, the maximum number of results for that worksheet, the source data, and transformation lists. All subsequent sheets need only to specify the polynomial order and the maximum number of results for that worksheet. The VBA subroutine copies the source data and transformation lists from the first worksheet to all other worksheets.

Figures 3 and 4 show the left and right sides, respectively, of a sample scheme P2 worksheet. This is the first sheet that also includes the source data, and transformation lists. Notice the following individual cells and columns of cells that provide input for the regression program:

- Cell A2 contains the polynomial order.
- Cell A4 contains the values for the maximum number of best models. The current value is 20.
- Column B has the values of the observed dependent variable Y, starting at cell B2.
- Column C stores the values of the observed independent variable X, starting at cell C2.
- Column D MUST BE empty.
- Column E contained the numerically coded transformations for variable Y.
- Column F contained the numerically coded transformations for variable X.
- Column G MUST BE empty.

The following worksheet columns provide the regression output generated by running the VBA macro **FindBestPolyReg**. You need to prepare the headers manually:

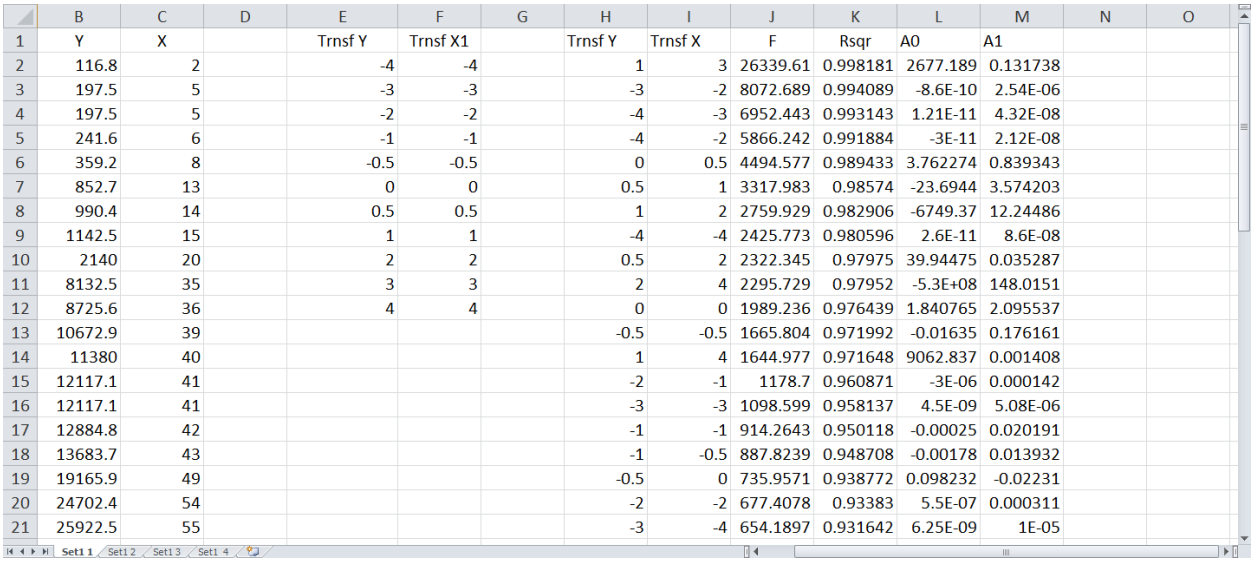
- Column H shows the best transformations for variable Y.
- Column I shows the best transformations for variable X.
- Column J displays the values of the F statistic.
- Column K shows the values of the R^2 statistic
- Column L displays the values of the intercepts.
- Column M shows the values for regression coefficient for the transformed variable raised to power 1.
- Columns N and up show values for regression coefficient for the transformed variable raised to power 2 and up, respectively.

The rows that start with the column H represent the record for each model. These rows are sorted in a descending order using the values of the F statistic (in column J).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Polynomial Order	Y	X		Trnsf Y	Trnsf X1		Trnsf Y	Trnsf X	F	Rsqr	A0	A1
2	1	116.8	2		-4	-4		1	3	26339.61	0.998181	2677.189	0.1317
3	Max Results	197.5	5		-3	-3		-3	-2	8072.689	0.994089	-8.6E-10	2.54E-(
4	20	197.5	5		-2	-2		-4	-3	6952.443	0.993143	1.21E-11	4.32E-(
5	Read comment	241.6	6		-1	-1		-4	-2	5866.242	0.991884	-3E-11	2.12E-(
6		359.2	8		-0.5	-0.5		0	0.5	4494.577	0.989433	3.762274	0.8393
7		852.7	13		0	0		0.5	1	3317.983	0.98574	-23.6944	3.5742(
8		990.4	14		0.5	0.5		1	2	2759.929	0.982906	-6749.37	12.244
9		1142.5	15		1	1		-4	-4	2425.773	0.980596	2.6E-11	8.6E-(
10		2140	20		2	2		0.5	2	2322.345	0.97975	39.94475	0.0352
11		8132.5	35		3	3		2	4	2295.729	0.97952	-5.3E+08	148.01
12		8725.6	36		4	4		0	0	1989.236	0.976439	1.840765	2.0955
13		10672.9	39					-0.5	-0.5	1665.804	0.971992	-0.01635	0.1761
14		11380	40					1	4	1644.977	0.971648	9062.837	0.0014(
15		12117.1	41					-2	-1	1178.7	0.960871	-3E-06	0.0001
16		12117.1	41					-3	-3	1098.599	0.958137	4.5E-09	5.08E-(
17		12884.8	42					-1	-1	914.2643	0.950118	-0.00025	0.0201
18		13683.7	43					-1	-0.5	887.8239	0.948708	-0.00178	0.0139
19		19165.9	49					-0.5	0	735.9571	0.938772	0.098232	-0.022
20		24702.4	54					-2	-2	677.4078	0.93383	5.5E-07	0.0003
21		25922.5	55					-3	-4	654.1897	0.931642	6.25E-09	1E-(

Set1 1 Set1 2 Set1 3 Set1 4

Figure 3. The left side of worksheet for scheme P2.



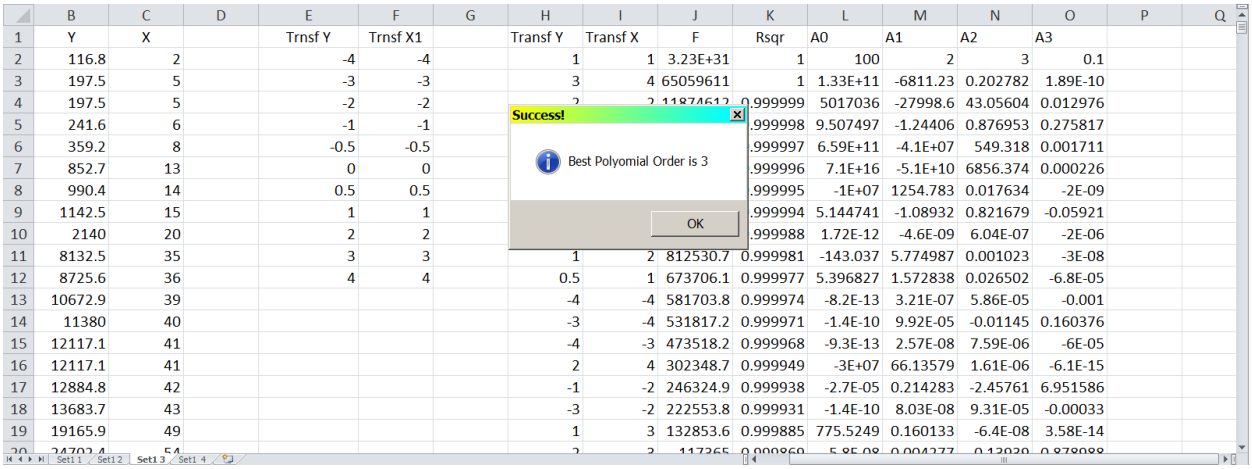
	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Y	X		Trnsf Y	Trnsf X1		Trnsf Y	Trnsf X	F	Rsqr	A0	A1		
2	116.8	2		-4	-4		1	3	26339.61	0.998181	2677.189	0.131738		
3	197.5	5		-3	-3		-3	-2	8072.689	0.994089	-8.6E-10	2.54E-06		
4	197.5	5		-2	-2		-4	-3	6952.443	0.993143	1.21E-11	4.32E-08		
5	241.6	6		-1	-1		-4	-2	5866.242	0.991884	-3E-11	2.12E-08		
6	359.2	8		-0.5	-0.5		0	0.5	4494.577	0.989433	3.762274	0.839343		
7	852.7	13		0	0		0.5	1	3317.983	0.98574	-23.6944	3.574203		
8	990.4	14		0.5	0.5		1	2	2759.929	0.982906	-6749.37	12.24486		
9	1142.5	15		1	1		-4	-4	2425.773	0.980596	2.6E-11	8.6E-08		
10	2140	20		2	2		0.5	2	2322.345	0.97975	39.94475	0.035287		
11	8132.5	35		3	3		2	4	2295.729	0.97952	-5.3E+08	148.0151		
12	8725.6	36		4	4		0	0	1989.236	0.976439	1.840765	2.095537		
13	10672.9	39					-0.5	-0.5	1665.804	0.971992	-0.01635	0.176161		
14	11380	40					1	4	1644.977	0.971648	9062.837	0.001408		
15	12117.1	41					-2	-1	1178.7	0.960871	-3E-06	0.000142		
16	12117.1	41					-3	-3	1098.599	0.958137	4.5E-09	5.08E-06		
17	12884.8	42					-1	-1	914.2643	0.950118	-0.00025	0.020191		
18	13683.7	43					-1	-0.5	887.8239	0.948708	-0.00178	0.013932		
19	19165.9	49					-0.5	0	735.9571	0.938772	0.098232	-0.02231		
20	24702.4	54					-2	-2	677.4078	0.93383	5.5E-07	0.000311		
21	25922.5	55					-3	-4	654.1897	0.931642	6.25E-09	1E-05		

Figure 4. The right side of worksheet for scheme P2.

The data in Figure 3 is based on the following polynomial:

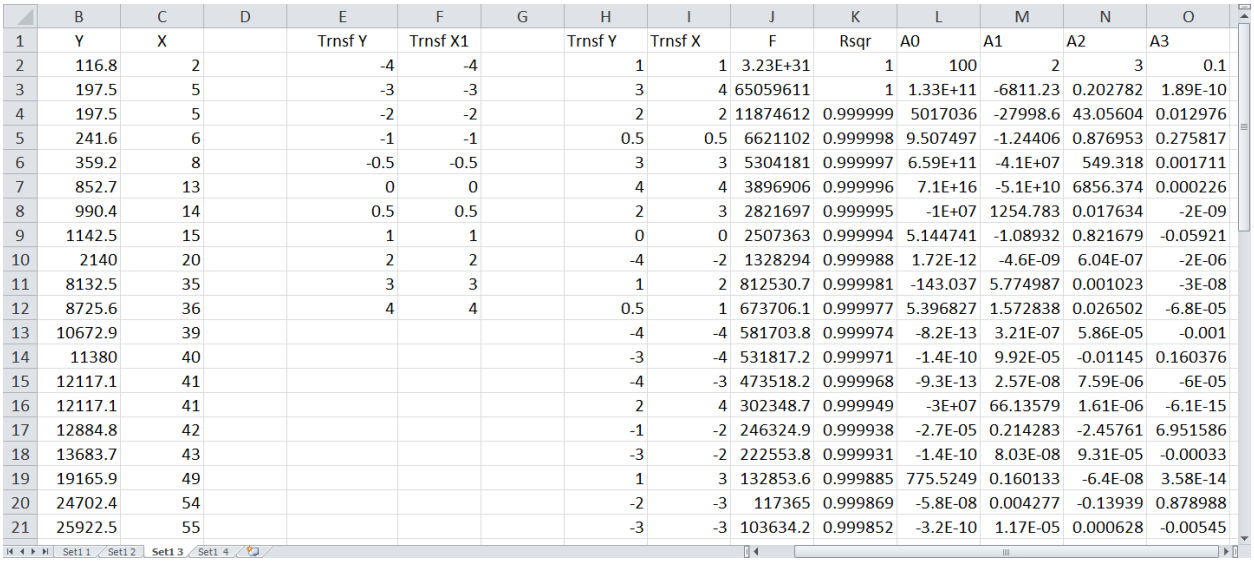
$$Y = 100 + 2\,X + 3\,X^2 + 0.1\,X^3$$

The workbook has four worksheets to perform polynomial regression in the orders 1 through 4. When you run the **FindBestPolyReg** subroutine, it selects worksheet Set 3 which has the cubic polynomial. Figures 5 and 6 show the best results of running the **FindBestPolyReg** subroutine.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Y	X		Trnsf Y	Trnsf X1		Transf Y	Transf X	F	Rsqr	A0	A1	A2	A3		
2	116.8	2		-4	-4		1	1	3.23E+31	1	100	2	3	0.1		
3	197.5	5		-3	-3		3	4	65059611	1	1.33E+11	-6811.23	0.202782	1.89E-10		
4	197.5	5		-2	-2		2	2	11874612	0.999999	5017036	-27998.6	43.05604	0.012976		
5	241.6	6		-1	-1					.999998	9.507497	-1.24406	0.876953	0.275817		
6	359.2	8		-0.5	-0.5					.999997	6.59E+11	-4.1E+07	549.318	0.001711		
7	852.7	13		0	0					.999996	7.1E+16	-5.1E+10	6856.374	0.000226		
8	990.4	14		0.5	0.5					.999995	-1E+07	1254.783	0.017634	-2E-09		
9	1142.5	15		1	1					.999994	5.144741	-1.08932	0.821679	-0.05921		
10	2140	20		2	2					.999988	1.72E-12	-4.6E-09	6.04E-07	-2E-06		
11	8132.5	35		3	3		1	2	812530.7	0.999981	-143.037	5.774987	0.001023	-3E-08		
12	8725.6	36		4	4		0.5	1	673706.1	0.999977	5.396827	1.572838	0.026502	-6.8E-05		
13	10672.9	39					-4	-4	581703.8	0.999974	-8.2E-13	3.21E-07	5.86E-05	-0.001		
14	11380	40					-3	-4	531817.2	0.999971	-1.4E-10	9.92E-05	-0.01145	0.160376		
15	12117.1	41					-4	-3	473518.2	0.999968	-9.3E-13	2.57E-08	7.59E-06	-6E-05		
16	12117.1	41					2	4	302348.7	0.999949	-3E+07	66.13579	1.61E-06	-6.1E-15		
17	12884.8	42					-1	-2	246324.9	0.999938	-2.7E-05	0.214283	-2.45761	6.951586		
18	13683.7	43					-3	-2	222553.8	0.999931	-1.4E-10	8.03E-08	9.31E-05	-0.00033		
19	19165.9	49					1	3	132853.6	0.999885	775.5249	0.160133	-6.4E-08	3.58E-14		
20	24702.4	54					2	3	117365	0.999869	5.8E-08	0.004277	0.13939	0.878988		

Figure 5. The message box that appears when subroutine FindBestPolyReg terminates.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Y	X		Trnsf Y	Trnsf X1		Trnsf Y	Trnsf X	F	Rsqr	A0	A1	A2	A3
2	116.8	2		-4	-4		1	1	3.23E+31	1	100	2	3	0.1
3	197.5	5		-3	-3		3	4	65059611	1	1.33E+11	-6811.23	0.202782	1.89E-10
4	197.5	5		-2	-2		2	2	11874612	0.999999	5017036	-27998.6	43.05604	0.012976
5	241.6	6		-1	-1		0.5	0.5	6621102	0.999998	9.507497	-1.24406	0.876953	0.275817
6	359.2	8		-0.5	-0.5		3	3	5304181	0.999997	6.59E+11	-4.1E+07	549.318	0.001711
7	852.7	13		0	0		4	4	3896906	0.999996	7.1E+16	-5.1E+10	6856.374	0.000226
8	990.4	14		0.5	0.5		2	3	2821697	0.999995	-1E+07	1254.783	0.017634	-2E-09
9	1142.5	15		1	1		0	0	2507363	0.999994	5.144741	-1.08932	0.821679	-0.05921
10	2140	20		2	2		-4	-2	1328294	0.999988	1.72E-12	-4.6E-09	6.04E-07	-2E-06
11	8132.5	35		3	3		1	2	812530.7	0.999981	-143.037	5.774987	0.001023	-3E-08
12	8725.6	36		4	4		0.5	1	673706.1	0.999977	5.396827	1.572838	0.026502	-6.8E-05
13	10672.9	39					-4	-4	581703.8	0.999974	-8.2E-13	3.21E-07	5.86E-05	-0.001
14	11380	40					-3	-4	531817.2	0.999971	-1.4E-10	9.92E-05	-0.01145	0.160376
15	12117.1	41					-4	-3	473518.2	0.999968	-9.3E-13	2.57E-08	7.59E-06	-6E-05
16	12117.1	41					2	4	302348.7	0.999949	-3E+07	66.13579	1.61E-06	-6.1E-15
17	12884.8	42					-1	-2	246324.9	0.999938	-2.7E-05	0.214283	-2.45761	6.951586
18	13683.7	43					-3	-2	222553.8	0.999931	-1.4E-10	8.03E-08	9.31E-05	-0.00033
19	19165.9	49					1	3	132853.6	0.999885	775.5249	0.160133	-6.4E-08	3.58E-14
20	24702.4	54					-2	-3	117365	0.999869	-5.8E-08	0.004277	-0.13939	0.878988
21	25922.5	55					-3	-3	103634.2	0.999852	-3.2E-10	1.17E-05	0.000628	-0.00545

Figure 6. The best results of running the FindBestPolyReg subroutine.

The best fitted polynomial has the F statistic of 1.238E+31 and the R^2 value of 1. The calculations identify the best polynomial as:

$$Y = 100 + 2\,X + 3\,X^2 + 0.1\,X^3$$

The FindBestPolyReg VBA Subroutine

Listing 2 shows the source code for subroutine **FindBestPolyReg**. This subroutine performs the following general tasks:

- Initializes the index of the best worksheet and F statistics for the best polynomial.
- Loops over the worksheets in the workbook.
  - Select the next worksheet.
  - If the current worksheet is not the first one, copy the data in columns B to columns K from the first worksheet to the current worksheet.
  - Invoke subroutine **BestPolyReg** to find the best transformations for the currently selected polynomial order. This code for the accompanying **BestPolyReg** subroutine is the same one used in scheme P1.
  - Store the best F and the index of the current worksheet if it provides the best F statistic.
- Select the worksheet that has the best polynomial model.
- Display a message box that states the order of the best polynomial model (see Figure 5 for a sample message box).

```
Option Explicit
Option Base 1
```



```
Sub FindBestPolyReg()
  Dim I As Integer, N As Integer
  Dim BestF As Double, BestOrder As Integer, BestSheet As Integer

  BestF = 0
  BestOrder = 0
  N = Application.Worksheets.Count
  For I = 1 To N
    Sheets(I).Select
    If I > 1 Then
      ' copy (X,Y) data and transformations from first sheet
      Sheets(1).Range("B1:K32000").Copy Range("B1:K32000")
    End If
    Call BestPolyReg
    If BestF < Cells(2, 10) Then
      BestF = Cells(2, 10)
      BestOrder = Cells(2, 1)
      BestSheet = I
    End If
  Next I
  Sheets(BestSheet).Select
  MsgBox "Best Polyomial Order is " & BestOrder, vbOKOnly +
vbInformation, "Success!"
End Sub

Sub BestPolyReg()

  Const MAX_ERRORS As Double = 1000000# ' initial max error value

  Dim ErrorCounter As Double, MaxErrors As Double, sMaxErr As String
  Dim NumIndpVars As Integer ' number of variables
  Dim PolyOrder As Integer ' polynomial order
  Dim TN As Integer ' total number of coefficients
  Dim N As Integer ' number of data points
  Dim MaxTrans As Integer ' max transformations
  Dim MaxResults As Integer ' max results
  Dim Col1 As Integer, Col2 As Integer, Col3 As Integer
  Dim Col4 As Integer, Col5 As Integer
  Dim I As Integer, J As Integer, K As Integer
  Dim M1 As Integer, M As Integer
  Dim iY As Integer, iX As Integer
  Dim TransfMat() As Double
  Dim CurrentTransf() As Double, CountTransf() As Integer
  Dim NumTransf() As Integer ' number of transformations
  Dim Y() As Variant, X() As Variant
```



```
  Dim Yt() As Variant, Xt() As Variant
  Dim Xtp() As Variant
  Dim vRegResultsMat As Variant
  Dim F As Double, Rsqr As Double, xval As Double

  ErrorCounter = 0
  MaxErrors = MAX_ERRORS
  NumIndpVars = 2
  PolyOrder = [A2].Value
  MaxResults = [A4].Value
  TN = PolyOrder + 1
  Col1 = 2              ' first column of data
  Col2 = 5  ' first column of transformations
  Col3 = 8  ' first column of best tranformations
  Col4 = 10 ' first column of results
  Col5 = Col4 + TN + 1  ' last column of results

  Range(Cells(1 + Col3), Cells(1, 50)).Value = ""
  Cells(1, Col3) = "Transf Y"
  Cells(1, Col3 + 1) = "Transf X"
  Cells(1, Col4) = "F"
  Cells(1, Col4 + 1) = "Rsqr"
  For I = 0 To PolyOrder
   Cells(1, Col4 + 2 + I) = "A" & I
  Next I

  Range(Cells(2, Col3), Cells(1 + 2 * MaxResults, Col4 + 3 *
TN)).Value = ""
  Range(Cells(2, Col4), Cells(1 + MaxResults, Col4 + 1 + TN)).Value =
0
  MaxTrans = Range(Cells(2, Col2), Cells(1,
Col2)).CurrentRegion.Rows.Count - 1
  ReDim NumTransf(NumIndpVars), TransfMat(NumIndpVars, MaxTrans),
CurrentTransf(NumIndpVars)
  ReDim CountTransf(NumIndpVars)

  For I = 1 To NumIndpVars
    J = 2
    Do While Trim(Cells(J, Col2 + I - 1)) <> ""
      TransfMat(I, J - 1) = Cells(J, Col2 + I - 1)
      J = J + 1
    Loop
    NumTransf(I) = J - 2
  Next I
```



```
  N = Range("B1").CurrentRegion.Rows.Count - 1
  Y = Range("B2:B" & N + 1).Value
  X = Range(Cells(2, 3), Cells(N + 1, 2 + PolyOrder)).Value
  Yt = Range("B2:B" & N + 1).Value
  Xt = Range(Cells(2, 3), Cells(N + 1, 2 + PolyOrder)).Value
  ReDim Xtp(N, PolyOrder) ' polynomial data matrix

  For iY = 1 To NumTransf(1)
    CurrentTransf(1) = TransfMat(1, iY)
    For iX = 1 To NumTransf(2)
      CurrentTransf(2) = TransfMat(2, iX)

      On Error GoTo HandleErr

      For I = 1 To N
        DoEvents

        If CurrentTransf(1) <> 0 Then
          Yt(I, 1) = Y(I, 1) ^ CurrentTransf(1)
        Else
          Yt(I, 1) = Log(Y(I, 1))
        End If

        If CurrentTransf(2) <> 0 Then
          Xt(I, 1) = X(I, 1) ^ CurrentTransf(2)
        Else
          Xt(I, 1) = Log(X(I, 1))
        End If

        For J = 1 To PolyOrder
          Xtp(I, J) = Xt(I, 1) ^ J
        Next J

      Next I

      ' perform the regression calculations
      vRegResultsMat = Application.WorksheetFunction.LinEst(Yt, Xtp,
True, True)

      Rsqr = vRegResultsMat(3, 1)
      F = vRegResultsMat(4, 1)
      ' check if F > F of last result
      If F > Cells(MaxResults + 1, Col4) Then
        M1 = MaxResults + 1
        ' write new results to row M
```



```
        Cells(M1, Col4) = F
        Cells(M1, Col4 + 1) = Rsqr
        For I = 1 To TN
          Cells(M1, Col4 + I + 1) = vRegResultsMat(1, TN - I + 1)
        Next I
        For I = 1 To NumIndpVars
          Cells(M1, Col3 + I - 1) = CurrentTransf(I)
        Next I

        Range(Cells(2, Col3), Cells(MaxResults + 1, Col5)).Select
        Range(Cells(2, Col3), Cells(MaxResults + 1, Col5)).Sort
Key1:=Range(Cells(2, Col4), Cells(MaxResults + 1, Col4)),
Order1:=xlDescending

      End If ' If F > Cells(MaxResults + 1, Col3)

    GoTo Here

HandleErr:
  ErrorCounter = ErrorCounter + 1
  If ErrorCounter > MaxErrors Then
    If MsgBox("Reached maximum error limits of " & ErrorCounter &
vbCrLf & _
    "Want to stop the process?", vbYesNo + vbQuestion, "Confirmation
requested") = vbYes Then
      Exit Sub
    Else
      sMaxErr = InputBox("Update maximum number of errors", "Max
Errors Input", MaxErrors)
      If Trim(sMaxErr) = "" Then
        MsgBox "User canceled calculations process", vbOKOnly +
vbInformation, "End of Process"
        Exit Sub
      End If
      MaxErrors = CDbl(sMaxErr)
      ErrorCounter = 0
    End If
  End If
  Resume Here
Here:
    Next iX
  Next iY
End Sub
```

Listing 2. The FindBestPolyReg subroutine.

References

1. Shammas, Namir, *Multiple Regression Model Using Excel Linest Function, article on www.namirshammas.com web site.*
2. Shammas, Namir, *Best Multiple Regression Model Using Excel Linest Function, article on www.namirshammas.com web site.*
3. Wikipedia article *Coefficient of Determination*.
4. Wikipedia article *Linear Regression*.
5. Wikipedia article *Simple Linear Regression*.
6. Wikipedia article *Akaike information criterion*.
7. Draper and Smith, *Applied Regression Analysis*, Wiley-Interscience; 3rd edition (April 23, 1998).
8. Neter, Kuther, Wasserman, and Nachtsheim, *Applied Linear Statistical Models*, McGraw-Hill/Irwin; 4th edition (February 1, 1996).
9. Fox, *Applied Regression Analysis and Generalized Linear Models*, Sage Publications, Inc; 2nd edition (April 16, 2008).
10. Montgomery, Peck, and Vining, *Introduction to Linear Regression Analysis*, Wiley-Interscience; 4th edition (2006).
11. Seber and Lee, *Linear Regression Analysis*, Wiley; 2nd edition (February 5, 2003).

About this Article

Version	*Date*	*Comment*
1.0.0.0	12/7/2012	Initial release.

