

Non-Linear Models

Mohammad Ehsanul Karim <wildscop@yahoo.com>
Institute of Statistical Research and training;
University of Dhaka, Dhaka – 1000, Bangladesh

Topic Outlines
Intrinsically Linear Regression Models
Intrinsically Non-linear Regression Models
Least Squares in the Non-linear Case
Linearization Method to obtain Estimates
Using Statistical Packages (SAS, R, NLREG)

Linear in parameters models (including only first order models in $p-1$ independent variables with p parameters) can be generalized as –

$$Y = b_0 + b_1 Z_1 + b_2 Z_2 + b_3 Z_3 + \dots + b_{p-1} Z_{p-1} + e$$

where the Z_i can represent any functions of the basic predictor variables $X_1, X_2, X_3, \dots, X_k$. While the above equation can represent a wide variety of relationships, there are many situations in which a model of this form is not appropriate; for example, when definite information is available about the form of the relationship between the response and the predictor variables. Such information might occasionally involve direct knowledge of the actual form of the true model or might be represented by a set of differential equations that the model must satisfy. Sometimes the information leads to several alternative models, may be of a non-linear form and we would usually prefer such a model whenever possible, rather than to fit an alternative, perhaps less realistic linear model^I.

A regression model is called nonlinear, if the derivatives of the model with respect to the model parameters depends on one or more parameters. This definition is essential to distinguish nonlinear from curvilinear regression. A regression model is not necessarily nonlinear if the graphed regression trend is curved. A polynomial model appears curved when y is plotted against x . It is, however, not a nonlinear model.

Fitting a nonlinear regression model to data is slightly more involved than fitting a linear model, but they have specific advantages:

- Nonlinear models are often derived on the basis of physical and/or biological considerations, e.g., from differential equations, and have justification within a quantitative conceptualization of the process of interest.
- The parameters of a nonlinear model usually have direct interpretation in terms of the process under study.
- Constraints can be built into a nonlinear model easily and are harder to enforce for linear models.

Intrinsically Linear and Intrinsically Non-linear Regression Models

Any model not of the above given form will be called a non-linear model, that is non-linear in parameters. Non-linear regression models can be classified into two groups according to whether they can or cannot be made linear with respect to the parameters to be estimated.

- 1. Intrinsically Linear Models:** A non-linear model with respect to the variables but linear with respect to the parameters to be estimated. Suitable transformations of data can frequently (not always) be found that will reduce a theoretical non-linear model to a linear form. These transformations are said to be “Linearizable^{II},” and comprise a class of functions that may exist either occur in practice or may themselves provide reasonable approximations to functions that occur in practice. Linearizing may require transforming both the independent and dependent variable. For instance, $Y = e^{(q_1 + q_2 t + e)}$ which can be transformed in a model which is

^I Although this is true that Linear regression models can provide a large and rich framework that suits the needs of many analysis, and the usefulness of linear models is more general than apparent, but linear regression cannot be adequate for all problems theoretically or empirically, since sometimes the response and the predictors are related through a known non-linear function.

^{II} This term is used by Chatterjee, Price (1991) “Regression analysis by Example”, 2nd ed., ch – 2 and Weisberg (1980) “Applied linear regression”, ch – 6.

linear in parameter just by taking logarithms to the base e – and we shall call the model Intrinsically Linear – that is, an intrinsically linear model is one that can be made linear by a transformation of parameters. The basic common characteristic of such models is that they can be converted into ordinary linear models by suitable transformation of variables – and such transformation amounts to nothing more than re-labeling one or more of the variables. While it may be useful, at times, to transform a model of this type so that it can be easily fitted, it will remain a non-linear model, whatever the transformation applied. However, the least squares estimates of the parameters will not in general be equivalent to the non-linear parameter estimates in the original model in this case. The reason is that – in the original non-linear model least squares implies minimization of the sum of squared residuals on y , whereas in the transformed model, we are minimizing the sum of squared residual on “transformed response” (that is, $\ln y$ in our present case). Also, the issue often resolves around the error structure^{III} (such as – so the standard assumptions on the errors apply to the original non-linear model or to the linearized one?) is sometimes not an easy question to answer.

- 2. Intrinsically Non-linear Models:** Not all functions are linearizable, nor in some cases it is desirable to transform to linearity. For example,

$$Y = \frac{q_1}{q_1 - q_2} [e^{-q_2 t} - e^{-q_1 t}] + e$$

however, is impossible to convert into a form – linear in parameters, which we will call Intrinsically Nonlinear^{IV}.

^{III} When an intrinsically linear model has been transformed into the linear model form, it is important to study the linearized model for aptness – since the normally distributed error term may not be normally distributed anymore when transformed.

^{IV} Nonlinear least squares and maximum log-likelihood are the most common methods for parameter estimation of nonlinear models in econometrics. For some more examples related to Economics, readers may consult Gujarati (2003) “Basic Econometrics” 4th ed., page – 564-5, Koutsoyiannis (1977) “Theory of Econometrics – An introduction exposition of Economic Methods”, 2nd ed., page – 134-8, Jan Kmenta (1986) “Elements of Econometrics”, 2nd ed., page – 503-26 for these two classifications.

Least Squares in the Non-linear Case

Notations: The standard notation for non-linear Least Squares^V situations is a bit different from that of linear Least Squares cases. Suppose the postulated model is of the form –

$$Y = f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k; \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p) + \mathbf{e}$$

where the \mathbf{q} 's are the parameters, and \mathbf{x} 's are the predictor variables and the \mathbf{e} term follows usual assumptions regarding error terms as we know from linear regression – such as $E(\mathbf{e}) = 0$ and $V(\mathbf{e}) = \mathbf{S}^2$, where $\mathbf{e} \sim N(0, \mathbf{S}^2)$ i.i.d and are uncorrelated..

If we write –

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_k \end{pmatrix}_{k \times 1} \quad \text{and} \quad \mathbf{q} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \dots \\ \mathbf{q}_p \end{pmatrix}_{p \times 1}$$

We denote the number of independent \mathbf{x} variables by k , since the number of \mathbf{x} variables in non-linear regression is not directly related to the number of parameters, unlike linear regression^{VI}.

Thus, the above model can be shorten to –

$$Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e}$$

$$\Rightarrow E(Y) = f(\mathbf{x}, \mathbf{q})$$

Now, let us extend the model for n observations as –

$$Y = f(\mathbf{x}_{1u}, \mathbf{x}_{2u}, \dots, \mathbf{x}_{ku}; \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p) + \mathbf{e}_u, \quad \text{for } u = (1, 2, \dots, n)$$

$$\Rightarrow Y_u = f(\mathbf{x}_u, \mathbf{q}) + \mathbf{e}_u$$

$$\text{where, } \mathbf{e} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \dots \\ \mathbf{e}_n \end{pmatrix} \sim N(0, I\mathbf{S}^2)$$

Procedure: We define the Error Sum of Squares^{VII} for the non-linear model as –

$$S(\mathbf{q}) = \sum_{u=1}^n [Y_u - f(\mathbf{x}_u, \mathbf{q})]^2$$

We shall denote by $\hat{\mathbf{q}}^{\text{VIII}}$, a least squares estimate of \mathbf{q} , that is a value of \mathbf{q} which minimizes $S(\mathbf{q})$. To find the least squares estimate $\hat{\mathbf{q}}$, we need to differentiate the Error Sum of Squares with respect to \mathbf{q} –

^V Ordinary Least squares applied to a non-linear regression model is called non-linear Least Squares.

^{VI} Readers may consult Neter, Wasserman, Kutner (1983) “Applied Linear Regression Models”, page – 470-1 for this matter.

^{VII} Since Y_u, \mathbf{x}_u are fixed observations, the sum of squares is a function of \mathbf{q} only.

^{VIII} It turns out that $\hat{\mathbf{q}}$ is also a Maximum Likelihood estimate of \mathbf{q} in this case.

$$\frac{\partial S(\mathbf{q})}{\partial \mathbf{q}_i} = \sum_{u=1}^n (-2) \{Y_u - f(\mathbf{x}_u, \mathbf{q})\} \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_i} \right]$$

When the p partial derivatives are set equal to zero, and the parameters \mathbf{q}_i are replaced by $\hat{\mathbf{q}}$, and after simplification, we obtain the p normal equations. The normal equations take the form –

$$\sum_{u=1}^n \{Y_u - f(\mathbf{x}_u, \mathbf{q})\} \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_i} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = 0, \text{ for } i = (1, 2, \dots, p)$$

where the quantity in the brackets is the derivative of $f(\mathbf{x}_u, \mathbf{q})$ with respect to \mathbf{q}_i with all \mathbf{q} 's replaced by the corresponding $\hat{\mathbf{q}}$'s, which have the same subscript. When the function $f(\mathbf{x}_u, \mathbf{q})$ was linear, this quantity was a function of \mathbf{x}_u only and did not involve the $\hat{\mathbf{q}}$'s at all. When the model is non-linear in the \mathbf{q} 's, so will be the normal equations.

Problem: However, in non-linear case, the solving the normal equations is not easy (the development of the least squares estimators for a non-linear model brings about complications not encountered in the case of the linear model) – and in most cases, it is extremely difficult to obtain – and iterative methods must be employed in nearly all cases. To compound the difficulties, it may happen that multiple solutions exist, corresponding to multiple stationary values of the function $S(\hat{\mathbf{q}})$. The statistical literature is quite rich in algorithms for minimization of the residual sum of squares in non-linear situations. Next, we will, therefore, discuss methods that have been used to estimate the parameters in non-linear systems.

Approaches to Estimation Non-linear Regression Models: In some non-linear problem, it is most convenient to write down the normal equations and develop a direct (method 1 below) iterative technique for solving them. Whether this works satisfactorily or not – depends on the form of the equations and the iterative method used. There are some of the alternative approaches –

1. Direct search (Trial-and-Error or Derivative-free technique^{IX})
2. Linearization (iterative method or Gauss-Newton Method)
3. Steepest descent^X (Direct Optimization)
4. Marquardt's compromise^{XI}

^{IX} This method is not generally used since it takes huge time and effort and; at the end of the day we may find some estimated parameter values with poor properties!

^X This method proceeds very systematically (and thus sometimes very slowly) starting with some initial values. This method is particularly effective when the starting values are not good and far from the final values. While, theoretically, the steepest descent method will converge, it may do so in practice with agonizing slowness after some rapid initial progress. A further disadvantage of the steepest descent method is that it is not scale invariant. The indicated direction of movement changes if the scales of the variables are changed, unless all are changed by the same factor. The steepest descent method is, on the whole, slightly less favored than the linearization method (described later) but will work satisfactorily for many nonlinear problems, especially if modifications are made to the basic technique.

^{XI} The method developed by D. W. Marquardt ("An algorithm for least squares estimation of nonlinear parameters", Journal of the Society for Industrial and Applied Mathematics, 2, 1963, 431-441) appears to enlarge considerably the number of practical problems that can be tackled by nonlinear estimation. This method is a compromise between the 2nd and 3rd method mentioned here, and occupies a middle ground between these two methods.

5. In addition to these approach, there are several currently employed methods available for obtaining the parameter estimates by Statistical Computer Packages

However, we will mention only the 2nd one in our present documentation.

Linearization Method: The Linearization^{XII} (followed by Gauss-Newton iteration based on Taylor Series^{XIII}) method uses the result of linear least squares in a succession of stages. Suppose the postulated model is of form:

$$Y_u = f(\mathbf{x}_u, \mathbf{q}) + \mathbf{e}_u$$

Let $\mathbf{q}_{10}, \mathbf{q}_{20}, \dots, \mathbf{q}_{p0}$ be initial values^{XIV} for the parameters $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p$. These initial values^{XV} may be guesses or preliminary estimates based on whatever information is available. These initial values will, hopefully, be improved upon in the successive iterations to be described below. If we carry out a Taylor series expansion of $f(\mathbf{x}_u, \mathbf{q})$ about the point \mathbf{q}_0 , where $\mathbf{q}_0 = (\mathbf{q}_{10}, \mathbf{q}_{20}, \dots, \mathbf{q}_{p0})'$ and curtail the expansion at the first derivatives, we can say that, approximately, when \mathbf{q} is close to \mathbf{q}_0 ,

^{XII} Linearization is accomplished by a Taylor series expansion of $f(\mathbf{x}_u, \mathbf{q})$ about the point (or initial starting vector values) with only the linear terms retained (ignoring all the higher order terms other than the first order term).

^{XIII} Any function (continuous and has a continuous p-th derivative) can be written as $f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \frac{(x-a)^3}{3!}f'''(a) + \dots + \frac{(x-a)^p}{p!}f^{(p)}(a) + R_{p+1}$

expanded by Taylor series where $f'(a) = df(x)/dx$ around $x = a$, and R_{p+1} is the remainder term – which is reasonably small if p is sufficiently large. Also, the function $f(x)$ must converge at the point a. If we omit the 2nd and higher order terms, then the equation is linear in “a” and we can apply a natural mechanism to estimate the coefficients in the linearized version of $f(x)$.

^{XIV} We also call them “Starting Values”. Sometimes, these are pure guesses, sometimes based on prior experience or prior empirical work or obtained by just fitting a linear regression model even though it may not be appropriate. All available prior information should be used to make these starting values as reliable as they possibly can be.

However, the choice of initial starting values is very important in this method because a poor choice may result in slow convergence. Good starting values will generally result in faster convergence, and if multiple maxima exist, will lead to a solution that is the global minimum rather than a local minimum. If there are several local minima in addition to an absolute minimum, poor starting values may result in convergence to an unwanted stationary point of the sum of squares surface. Also, it is often desirable to try other set of starting values after a solution has been obtained to make sure that the same solution will be found.

Readers may consult Myers (1986) “Classical and Modern Regression with Applications”, page 316-7, Draper, Smith (1998) “Applied Regression Analysis”, Third ed., Page 517-8; Montgomery, Peck (1992) “Introduction to Linear Regression Analysis”, 2nd ed., Page – 431-2 for leaning how they suggest to get such initial values.

‘Grid points’ in the parameter space is often useful. Some Statistical packages for non-linear requires that the users specify the starting values for the regression parameter, and others do a grid search to obtain starting values.

^{XV} For example, they may be values suggested by the information gained in fitting a similar equation in a different laboratory or suggested as “about right” by the experimenter based on his/her experience and knowledge.)

$$f(\mathbf{x}_u, \mathbf{q}) = f(\mathbf{x}_u, \mathbf{q}_0) + \sum_{i=1}^p \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_i} \right]_{\mathbf{q}=\mathbf{q}_0} (\mathbf{q}_i - \mathbf{q}_{i0})$$

$$f(\mathbf{x}_u, \mathbf{q}) = \left\{ \begin{array}{l} f(\mathbf{x}_u, \mathbf{q}_0) \\ + \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_1} \right]_{\mathbf{q}=\mathbf{q}_0} (\mathbf{q}_1 - \mathbf{q}_{10}) \\ + \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_2} \right]_{\mathbf{q}=\mathbf{q}_0} (\mathbf{q}_2 - \mathbf{q}_{20}) \\ + \dots + \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_p} \right]_{\mathbf{q}=\mathbf{q}_0} (\mathbf{q}_p - \mathbf{q}_{p0}) \end{array} \right\}$$

This above equation represents what is essentially a linearization of the non-linear form of the function $f(\mathbf{x}_u, \mathbf{q})$ - and the equation may be viewed as a linear approximation in a neighborhood of the starting values..

Now, we set –

$$f_u^0 = f(\mathbf{x}_u, \mathbf{q}_0),$$

$$\mathbf{b}_i^0 = (\mathbf{q}_i - \mathbf{q}_{i0}),$$

$$Z_{iu}^0 = \left[\frac{\partial f(\mathbf{x}_u, \mathbf{q})}{\partial \mathbf{q}_i} \right]_{\mathbf{q}=\mathbf{q}_0}$$

and get the new form –

$$Y_u = f_u^0 + \mathbf{b}_1^0 Z_{1u}^0 + \mathbf{b}_2^0 Z_{2u}^0 + \dots + \mathbf{b}_p^0 Z_{pu}^0 + \mathbf{e}_u = f_u^0 + \sum_{i=1}^p \mathbf{b}_i^0 Z_{iu}^0 + \mathbf{e}_u$$

$$\Rightarrow Y_u - f_u^0 = \sum_{i=1}^p \mathbf{b}_i^0 Z_{iu}^0 + \mathbf{e}_u$$

which is very similar to the linear regression model and the procedure essentially involves a linear regression analysis on the above model.

If we write –

$$Z_0 = \langle Z_{iu}^0 \rangle = \begin{bmatrix} Z_{11}^0 & Z_{21}^0 & \dots & Z_{p1}^0 \\ Z_{12}^0 & Z_{22}^0 & \dots & Z_{p2}^0 \\ \dots & \dots & \dots & \dots \\ Z_{1n}^0 & Z_{2n}^0 & \dots & Z_{pn}^0 \end{bmatrix}_{(n \times p)}, \text{ for } (i,u) = [(1,2,\dots,p), (1,2,\dots,n)]$$

$$\hat{\mathbf{b}}_0 = \mathbf{b}_0 = \begin{bmatrix} b_1^0 \\ b_2^0 \\ \dots \\ b_p^0 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}}_1^0 \\ \hat{\mathbf{b}}_2^0 \\ \dots \\ \hat{\mathbf{b}}_p^0 \end{bmatrix}_{p \times 1} \text{ and } y_0 = \langle Y - f^0 \rangle = \begin{bmatrix} Y_1 - f_1^0 \\ Y_2 - f_2^0 \\ \dots \\ Y_n - f_n^0 \end{bmatrix}_{n \times 1}$$

then the reformed model is –

$$y_0 = Z_0 \hat{\mathbf{b}}_0 + \mathbf{e} = Z_0 \mathbf{b}_0 + \mathbf{e}$$

and the estimate^{XVI} of \mathbf{b}_0 , b_0 is given by –

$$b_0 = (Z_0'Z_0)^{-1}Z_0'(Y - f^0) = \text{a } p \times 1 \text{ vector}$$

Thus, the vector b_0 will be minimizing the Sum of Squares –

$$SS(\mathbf{q}) = \sum_{u=1}^n \left[Y_u - f(\mathbf{x}_u, \mathbf{q}_0) - \sum_{i=1}^p \mathbf{b}_i^0 Z_{iu}^0 \right]^2$$

with respect to the \mathbf{b}_i^0 , where $\mathbf{b}_i^0 = \mathbf{q}_i - \hat{\mathbf{q}}_{i0}$. Note that the above equation is the approximating linear expansion of our model.

At this point, we can examine whether the revised regression coefficients represent adjustments in the proper direction by checking some criterion measure. If the method is working effectively, then $SSE^{(1)}$ should be smaller than $SSE^{(0)}$ since the revised regression coefficients should be better estimates^{XVII}.

Of course, the analyst cannot work with the approximated linear model directly in a one step operation (need to be iterated). Let us write $b_i^0 = \mathbf{q}_{i1} - \hat{\mathbf{q}}_{i0}$, then the \mathbf{q}_{i1} ; $i = 1, 2, \dots, p$ can be thought as the revised best estimate of \mathbf{q} . Now, we can replace the values \mathbf{q}_{i1} , the revised estimates, in the same roles as were played above by the values \mathbf{q}_{i0} and go exactly the same procedure described above, by replacing all zero subscripts by ones. This will lead to another set of revised estimates \mathbf{q}_{i2} , and so on.

In vector form, we can write about the things of j -th iteration as–

$$\hat{\mathbf{q}}_{j+1} = \hat{\mathbf{q}}_j + b_j$$

$$\Rightarrow \hat{\mathbf{q}}_{j+1} = \hat{\mathbf{q}}_j + (Z_j'Z_j)^{-1}Z_j'(Y - f^j)$$

$$\text{where, } Z_j = \langle Z_{iu}^j \rangle_{n \times p}, f^j = \begin{pmatrix} f_1^j \\ f_2^j \\ \dots \\ f_n^j \end{pmatrix}, \text{ and } \mathbf{q}_j = \begin{pmatrix} \mathbf{q}_{1j} \\ \mathbf{q}_{2j} \\ \dots \\ \mathbf{q}_{pj} \end{pmatrix}$$

^{XVI} Any analyst who embarks on a non-linear estimation exercise should be made aware of what is known (or unknown) about the properties of the estimators. In the non-linear case, we cannot make any general statements about the properties of the estimators except for large sample – since only approximate procedures for statistical tests and confidence intervals are available. The estimators are not unbiased in general, but they are unbiased and minimum variance estimators in the limit, that is, as the sample grows large. Therefore, non-linear estimates do not possess optimal properties in finite samples – thus, the results found from small samples must be interpreted carefully. There are asymptotic variance-covariance results that we can use to obtain approximate confidence intervals and to construct t-statistics on the parameters. For the asymptotic covariance matrix formula, readers may consult Montgomery, Peck (1992) “Introduction to Linear Regression Analysis”, 2nd ed., Page - 427.

^{XVII} For details, readers may consult Neter, Wasserman, Kutner (1983) “Applied Linear Regression Models”, page 474-9, and Myers (1986) “Classical and Modern Regression with Applications”, page 305-12.

In general, the exact iteration process is as follows:

1. Estimate \mathbf{b} 's in the model by linear least squares. We shall denote these first iteration estimates by \mathbf{b} 's.
2. Compute $\hat{\mathbf{q}}_{i1} = \hat{\mathbf{q}}_{i0} + \mathbf{b}_i$ ($i = 1, 2, \dots, p$) but this is not our final estimates.
3. The $\hat{\mathbf{q}}_{i1}$ value is treated as the initial value in our first approximated linear model.
4. We return to the first step and again compute \mathbf{b} 's (at each iteration, new \mathbf{b} 's represent 'increments' that are added to the estimates from the previous iteration according to 2nd step) and eventually find $\hat{\mathbf{q}}_{i2}$.
5. We continue the process until convergence^{XVIII} is reached.

Stopping Rule: This iteration process is continued until the solution converges, that is, until in successive iterations $j, (j+1)$,

$$\left| \frac{\{\mathbf{q}_{i(j+1)} - \mathbf{q}_{ij}\}}{\mathbf{q}_{ij}} \right| < \mathbf{d}, \quad i = (1, 2, \dots, p)$$

where \mathbf{d} is some predetermined small amount (e.g., 0.000001 or 1.0×10^{-6}). At each stage of the iteration procedure, $S(\mathbf{q}_i)$ can be evaluated to see if a reduction in its value has actually been achieved.

Limitations:

The user may experience some difficulties with the Linearization procedure, such as –

1. It may converge very slowly, that is, a very large number of iteration may be required before the solution stabilizes even though the sum of squares $S(\mathbf{q}_i)$ may decrease consistently as j increases. This sort of behavior is not common but occurs.
2. It may oscillate widely, continually reversing direction, and often increasing, as well as decreasing the sum of squares. Nevertheless, the solution may stabilize eventually.
3. It may not converge at all, and even diverge (move at the wrong direction), so that the sum of squares increases iteration after iteration without bound.

For these drawbacks, several modifications of this process has been suggested^{XIX} to improve its performance.

^{XVIII} Convergence implies that after, say r iterations, the residual sum of squares and the parameter estimates are no longer (significantly) changing.

^{XIX} Readers may consult Montgomery, Peck (1992) "Introduction to Linear Regression Analysis", 2nd ed., Page – 428; Draper, Smith (1998) "Applied Regression Analysis", Third ed., Page 510-1 for these.

Readers may like to consult Bates, D. M. and Watts, D. G. (1988) "Nonlinear Regression Analysis and Its Applications", New York: John Wiley and Sons (specially ch-2,3; page-32-133) for more general discussion (however, not best for the starters / beginners of regression) about this Non-linear Regression topic – which is very much appreciated by so many other authors of this topic. In fact, the popular S software's (origin of S-plus and R) methods are based on those described in Bates and Watts. Also, Seber, G. A. F. and Wild, C. J. (1989) "Nonlinear Regression", New York: John Wiley and Sons is another encyclopaedic reference.

Nonlinear Regression using Statistical Software: SAS

The SAS System offers a powerful procedure to fit nonlinear regression models, PROC NLIN^{XX}. The NLIN procedure implements iterative methods that attempt to find least-squares estimates for nonlinear models. This procedure performs univariate nonlinear regression using the least squares method. It was improved dramatically in release 6.12 of The SAS System with the addition of a differentiator. Prior to release 6.12, if we wanted to fit a nonlinear model we had to supply the model specification as well as the formulas for the derivatives of the model. The latter was a real hassle, especially if the model is complicated. A method to circumvent the specification of derivatives was to choose a fitting algorithm that approximates the derivatives by differences. This algorithm, known as DUD (Does not Use Derivatives) was hence very popular. However, the algorithm is also known to be quite poor in computing good estimates. A method using derivatives is to be preferred. With release 6.12 SAS will calculate derivatives for us if we wish. The user still has the option to supply derivatives. It is, however, recommended to let The SAS System calculate them for us. The minimum specification to fit a nonlinear regression with PROC NLIN demands that the user specify the model and the parameters in the model. All terms in the model not defined as parameters are looked for in the data set that PROC NLIN processes. Since nonlinear models are often difficult to estimate, PROC NLIN may not always find the globally optimal least-squares estimates.

Using a Starting Value Grid: A grid search is also available to select starting values for the parameters. If we are not sure about the starting values, we can use a grid by offering SAS more than one starting value. It will calculate the initial residual sum of squares for all combinations of starting values and start the iterations with the best set.

Choosing the Fitting Algorithm: If our data and model are well behaved, it should not make a difference how we fit the nonlinear model to data. Unfortunately, this can not be said for all nonlinear regression models. We may have to choose carefully, which algorithm to use. In PROC NLIN different fitting algorithms are invoked with the METHOD= option of the PROC NLIN statement. Here are a few guidelines:

- If possible, choose a method that uses derivatives, avoid DUD. Unfortunately, if we do not specify derivatives and a METHOD= option, SAS will default to the DUD method.
- If the parameters are highly correlated, choose the Levenberg-Marquardt method (keyword METHOD=MARQUARDT)
- Among the derivative dependent methods, prefer the Newton-Raphson (METHOD=NEWTON) over the Gauss (METHOD=GAUSS) method.

Calculating predicted values and their confidence intervals: Predicted values are not displayed on screen in PROC NLIN. However, we can request to save them to a data set for later use. Along with the predicted values, we can calculate confidence bounds for the mean predictions, prediction intervals for an individual predictions and so forth.

^{XX} I am indebted to Professor Oliver Schabenberger (1998) of Virginia Tech for producing this SAS part of this documentation. For more about this SAS procedure, readers may consult via internet at <http://www.id.unizh.ch/software/unix/statmath/sas/sasdoc/stat/chap45/sect2.htm>

Nonlinear Regression using Statistical Software: R

R uses `nls()` function to perform estimation in non-linear Regression models. Also, the same function exists in S-plus 4 and above. The arguments^{xxi} to `nls` are the following.

formula

A non-linear model formula. The form is `response ~ mean`, where the right-hand side can have either of two forms. The standard form is an ordinary algebraic expression containing both parameters and determining variables. Note that the operators now have their usual arithmetical meaning.

data

An optional data frame for the variables (and sometimes parameters).

start

A list or numeric vector specifying the starting values for the parameters in the model. The names of the components of `start` are also used to specify which of the variables occurring on the right-hand side of the model formula are parameters. All other variables are then assumed to be determining variables.

control

An optional argument allowing some features of the default iterative procedure to be changed.

algorithm

An optional character string argument allowing a particular fitting algorithm to be specified. The default procedure is simply "default".

trace

An argument allowing tracing information from the iterative procedure to be printed. By default none is printed.

^{xxi} I am indebted to W. N. Venables and B. D. Ripley (2002) "Modern Applied Statistics with S", 4th ed., for producing this part of the documentation.

Illustrative Example:

Let the data^{xxii} file is in “c:\ draper.txt”, which is as follows:

t	Y
1	0.80
4	0.45
16	0.04

Let that, we have to estimate the parameter q in the non-linear model $Y = e^{-qt} + e$ from the above observations.

Now for starting analysis, we need a “initial value”. But we have no prior information about what values q might take. There are a few suggestions for such situations: one of which is to make a grid search using Statistical Software. Let us demonstrate how we can do this.

Using Statistical Software: SAS: Making a Grid Search

We write the following program in SAS (I used SAS 6.12) to get an estimate (note that we told SAS to do the search in Linearization technique by setting METHOD=NEWTON):

```
DATA DS;
INPUT T Y;
DATALINES;
1 0.80
4 0.45
16 0.04
RUN;

PROC NLIN DATA=DS METHOD=NEWTON;
PARAMETERS G=-100 TO 100 BY .1;
MODEL Y = EXP(-(G*T));
OUTPUT OUT=NLINOUT PREDICTED=PRED L95M=L95MEAN
U95M=U95MEAN L95=L95IND U95=U95IND;
RUN;

PROC PRINT DATA=NLINOUT;
RUN;
```

There will be an enormous search to find this estimate since we assumed our potential parameter (in the program, we wrote $q = G$) space to be -100 to 100 and told SAS to look within each 0.1 interval (so that we can get minimum residual SS or minima). However, after some iteration (when a convergent value will be found) SAS will bring the following output (edited to shorten the length – but the key parts of the output is unedited):

^{xxii} We are using this simple and small data set just for easy illustration purpose. This data set is taken from Draper, Smith (1998) “Applied Regression Analysis”, Third ed., Page 553, Exercise 24.A.

The SAS System 11:52 Sunday,
(Detailed Results on Non-Linear Least Squares Grid Search is omitted from here.)

Non-Linear Least Squares Iterative Phase

Iter	Dependent Variable Y G	Method: Newton Sum of Squares
0	0.200000	0.000352
1	0.203367	0.000302
2	0.203449	0.000302
3	0.203449	0.000302

NOTE: Convergence criterion met.

The SAS System 11:52 Sunday, April 6, 1997 50

Non-Linear Least Squares Summary Statistics Dependent Variable Y

Source	DF	Sum of Squares	Mean Square
Regression	1	0.84379816630	0.84379816630
Residual	2	0.00030183370	0.00015091685
Uncorrected Total	3	0.84410000000	
(Corrected Total)	2	0.28940000000	

Parameter	Estimate	Asymptotic Std. Error	Asymptotic 95 % Confidence Interval Lower Upper
G	0.2034488738	0.00603799034	0.17746921275 0.22942853484

Asymptotic Correlation Matrix

Corr	G
1	1

OBS	T	Y	PRED	L95MEAN	U95MEAN	L95IND	U95IND
1	1	0.80	0.81591	0.79472	0.83711	0.75896	0.87286
2	4	0.45	0.44317	0.39712	0.48923	0.37307	0.51328
3	16	0.04	0.03857	0.02254	0.05461	-0.01666	0.09381

Therefore, from SAS, we get estimate of $\mathbf{q} = 0.2034488738$. So we may suppose that the actual value may be around zero for now and check it in other ways.

Calculation By Hand: Non-linear Regression:

We will now go through some iterations just to see how the process works. Readers may think it as a slow motion of what happens to any Statistical Software when we run a non-linear regression.

Iteration 0: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated initial value $\mathbf{q}_0 = 0$ (as our grid search indicates – note that, since we have to estimate only one value, this is just a scaler, but for multi-parameter case, we would have a vector).

$$f^0 = f(\mathbf{x}, \mathbf{q}_0) = e^{-0 \times t} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_0 = \langle Y - f^0 \rangle = \begin{bmatrix} Y_1 - f_1^0 \\ Y_2 - f_2^0 \\ Y_3 - f_3^0 \end{bmatrix} = \begin{bmatrix} 0.80 - 1 \\ 0.45 - 1 \\ 0.04 - 1 \end{bmatrix} = \begin{bmatrix} -0.20 \\ -0.55 \\ -0.96 \end{bmatrix}$$

$$Z_0 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\mathbf{q}_0} = (-t) \times e^{-0 \times t} = \begin{bmatrix} -1 \\ -4 \\ -16 \end{bmatrix}$$

$$Z_0' Z_0 = 1^2 + 4^2 + 16^2 = 273$$

$$(Z_0' Z_0)^{-1} = (273)^{-1} = 0.003663004$$

$$Z_0' (Y - f^0) = (-0.20) \times (-1) + (-0.55) \times (-4) + (-0.96) \times (-16) = 17.76$$

$$b_0 = (Z_0' Z_0)^{-1} Z_0' (Y - f^0) = 0.003663004 \times 17.76 = 0.06505495$$

$$\hat{\mathbf{q}}_1 = \hat{\mathbf{q}}_0 + b_0 = 0.06505495 + 0 = 0.06505495$$

Iteration 1: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_1 = 0.06505495$.

$$f^1 = f(\mathbf{x}, \mathbf{q}_1) = e^{-\hat{\mathbf{q}} \times t} = \begin{bmatrix} 0.9370160 \\ 0.7708821 \\ 0.3531441 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_1 = \langle Y - f^1 \rangle = \begin{bmatrix} Y_1 - f_1^1 \\ Y_2 - f_2^1 \\ Y_3 - f_3^1 \end{bmatrix} = \begin{bmatrix} -0.1370160 \\ -0.3208821 \\ -0.3131441 \end{bmatrix}$$

$$Z_1 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}} \times t} = \begin{bmatrix} -0.937016 \\ -3.083529 \\ -5.650305 \end{bmatrix}$$

$$Z_1' Z_1 = 42.3121$$

$$(Z_1' Z_1)^{-1} = (42.3121)^{-1}$$

$$Z_1' (Y - f^1) = 2.887195$$

$$b_1 = (Z_1' Z_1)^{-1} Z_1' (Y - f^1) = 0.06823569$$

$$\hat{\mathbf{q}}_2 = \hat{\mathbf{q}}_1 + b_1 = 0.1332906$$

Iteration 2: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_1 = 0.1332906$.

$$f^2 = f(\mathbf{x}, \mathbf{q}_2) = e^{-\hat{\mathbf{q}} \times t} = \begin{bmatrix} 0.8752107 \\ 0.5867464 \\ 0.1185228 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_2 = \langle Y - f^2 \rangle = \begin{bmatrix} Y_1 - f_1^2 \\ Y_2 - f_2^2 \\ Y_3 - f_3^2 \end{bmatrix} = \begin{bmatrix} -0.07521069 \\ -0.13674643 \\ -0.07852278 \end{bmatrix}$$

$$Z_2 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}}^t} = \begin{bmatrix} -0.8752107 \\ -2.3469857 \\ -1.8963644 \end{bmatrix}$$

$$Z_2' Z_2 = 9.870534$$

$$(Z_2' Z_2)^{-1} = (9.870534)^{-1}$$

$$Z_2' (Y - f^2) = 0.5356749$$

$$b_2 = (Z_2' Z_2)^{-1} Z_2' (Y - f^2) = 0.05427011$$

$$\hat{\mathbf{q}}_3 = \hat{\mathbf{q}}_2 + b_2 = 0.1875607$$

Iteration 3: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_3 = 0.1875607$.

$$f^3 = f(\mathbf{x}, \mathbf{q}_3) = e^{-\hat{\mathbf{q}}_3^t} = \begin{bmatrix} 0.82897877 \\ 0.47225180 \\ 0.04973871 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_3 = \langle Y - f^3 \rangle = \begin{bmatrix} Y_1 - f_1^3 \\ Y_2 - f_2^3 \\ Y_3 - f_3^3 \end{bmatrix} = \begin{bmatrix} -0.028978765 \\ -0.022251802 \\ -0.009738708 \end{bmatrix}$$

$$Z_3 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}}_3^t} = \begin{bmatrix} -0.8289788 \\ -1.8890072 \\ -0.7958193 \end{bmatrix}$$

$$Z_3' Z_3 = 4.888882$$

$$(Z_3' Z_3)^{-1} = (4.888882)^{-1}$$

$$Z_3' (Y - f^3) = 0.07380705$$

$$b_3 = (Z_3' Z_3)^{-1} Z_3' (Y - f^3) = 0.01509691$$

$$\hat{\mathbf{q}}_4 = \hat{\mathbf{q}}_3 + b_3 = 0.2026576$$

Iteration 4: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_4 = 0.2026576$

$$f^4 = f(\mathbf{x}, \mathbf{q}_4) = e^{-\hat{\mathbf{q}}_4^t} = \begin{bmatrix} 0.81655777 \\ 0.44457770 \\ 0.03906526 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_4 = \langle Y - f^4 \rangle = \begin{bmatrix} Y_1 - f_1^4 \\ Y_2 - f_2^4 \\ Y_3 - f_3^4 \end{bmatrix} = \begin{bmatrix} -0.0165577742 \\ 0.0054223029 \\ 0.0009347429 \end{bmatrix}$$

$$Z_4 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}}^t} = \begin{bmatrix} -0.8165578 \\ -1.7783108 \\ -0.6250441 \end{bmatrix}$$

$$Z_4' Z_4 = 4.219836$$

$$(Z_4' Z_4)^{-1} = (4.219836)^{-1}$$

$$Z_4' (Y - f^4) = 0.003293584$$

$$b_4 = (Z_4' Z_4)^{-1} Z_4' (Y - f^4) = 0.0007805005$$

$$\hat{\mathbf{q}}_5 = \hat{\mathbf{q}}_4 + b_4 = 0.2034381$$

Iteration 5: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_5 = 0.2034381$

$$f^5 = f(\mathbf{x}, \mathbf{q}_5) = e^{-\hat{\mathbf{q}}_5^t} = \begin{bmatrix} 0.81592070 \\ 0.44319189 \\ 0.03858044 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_5 = \langle Y - f^5 \rangle = \begin{bmatrix} Y_1 - f_1^5 \\ Y_2 - f_2^5 \\ Y_3 - f_3^5 \end{bmatrix} = \begin{bmatrix} -0.015920699 \\ 0.006808111 \\ 0.001419557 \end{bmatrix}$$

$$Z_5 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}}_5^t} = \begin{bmatrix} -0.8159207 \\ -1.7727676 \\ -0.6172871 \end{bmatrix}$$

$$Z_5' Z_5 = 4.189475$$

$$(Z_5' Z_5)^{-1} = (4.189475)^{-1}$$

$$Z_5' (Y - f^5) = 4.455581 \times 10^{-05}$$

$$b_5 = (Z_5' Z_5)^{-1} Z_5' (Y - f^5) = 1.063518 \times 10^{-05}$$

$$\hat{\mathbf{q}}_6 = \hat{\mathbf{q}}_5 + b_5 = 0.2034487$$

Iteration 6: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_6 = 0.2034487$

$$f^6 = f(\mathbf{x}, \mathbf{q}_6) = e^{-\hat{\mathbf{q}}_6^t} = \begin{bmatrix} 0.81591202 \\ 0.44317304 \\ 0.03857388 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_6 = \langle Y - f^6 \rangle = \begin{bmatrix} Y_1 - f_1^6 \\ Y_2 - f_2^6 \\ Y_3 - f_3^6 \end{bmatrix} = \begin{bmatrix} -0.015912022 \\ 0.006826964 \\ 0.001426121 \end{bmatrix}$$

$$Z_6 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}}^t} = \begin{bmatrix} -0.8159120 \\ -1.7726921 \\ -0.6171821 \end{bmatrix}$$

$$Z_6' Z_6 = 4.189064$$

$$(Z_6' Z_6)^{-1} = (4.189064)^{-1}$$

$$Z_6' (Y - f^6) = 5.276473 \times 10^{-07}$$

$$b_6 = (Z_6' Z_6)^{-1} Z_6' (Y - f^6) = 1.259583 \times 10^{-07}$$

$$\hat{\mathbf{q}}_7 = \hat{\mathbf{q}}_6 + b_6 = 0.2034489$$

Iteration 7: We have the function $Y = f(\mathbf{x}, \mathbf{q}) + \mathbf{e} = e^{-q^t} + \mathbf{e}$ in our hand with approximated value $\hat{\mathbf{q}}_7 = 0.2034489$

$$f^7 = f(\mathbf{x}, \mathbf{q}_7) = e^{-\hat{\mathbf{q}}_7^t} = \begin{bmatrix} 0.8159119 \\ 0.4431728 \\ 0.0385738 \end{bmatrix} \text{ since } t \text{ is a } 3 \times 1 \text{ vector.}$$

$$\text{Now, } y_7 = \langle Y - f^7 \rangle = \begin{bmatrix} Y_1 - f_1^7 \\ Y_2 - f_2^7 \\ Y_3 - f_3^7 \end{bmatrix} = \begin{bmatrix} -0.015911919 \\ 0.006827188 \\ 0.001426199 \end{bmatrix}$$

$$Z_7 = \left[\frac{\partial f(\mathbf{x}, \mathbf{q})}{\partial \mathbf{q}} \right]_{\mathbf{q}=\hat{\mathbf{q}}} = (-t) \times e^{-\hat{\mathbf{q}}_7^t} = \begin{bmatrix} -0.8159119 \\ -1.7726912 \\ -0.6171808 \end{bmatrix}$$

$$Z_7' Z_7 = 4.189059$$

$$(Z_7' Z_7)^{-1} = (4.189059)^{-1}$$

$$Z_7' (Y - f^7) = 6.236455 \times 10^{-09}$$

$$b_7 = (Z_7' Z_7)^{-1} Z_7' (Y - f^7) = 1.488749 \times 10^{-09}$$

$$\hat{\mathbf{q}}_8 = \hat{\mathbf{q}}_7 + b_7 = 0.2034489$$

Since $\hat{\mathbf{q}}_8 = \hat{\mathbf{q}}_7$ we terminate the process and declare $\hat{\mathbf{q}}_8 = 0.2034489$ as our desired estimate.

Using Statistical Software: R: Non-linear Regression

Now, using Statistical Software like R (I used R 1.7.1) we can do the whole procedure just in a blink (and of course, without pain!) by writing the following commands in the R console:

```
> options(prompt=" R> " )
R> draper<-read.table("c:\\draper.txt",header=T)
R> draper.data<-data.frame(draper)
R> attach(draper.data)
R> nls(Y~exp(-(g*t)), data=draper.data, start = c(g =
0), trace = T)

1.2641 : 0
0.2197979 : 0.06505495
0.03052205 : 0.1332906
0.001429753 : 0.1875607
0.000304435 : 0.2026576
0.0003018342 : 0.2034381
0.0003018337 : 0.2034487
0.0003018337 : 0.2034489
Nonlinear regression model
  model: Y ~ exp(-(g * t))
  data: draper.data
           g
0.2034489
residual sum-of-squares: 0.0003018337

R> coef(nls(Y~exp(-(g*t)), data=draper.data, start = c(g
= 0), trace = T))

(---)
           g
0.2034489
```

Note that, the estimated value of $q = 0.2034489$ approximately in all our (slow motion hand calculation and R results within a blink) cases – which matches our SAS result too. Thus, we declare 0.2034489 as our estimate of the parameter q based on the given data.

By now, readers may think that R is not capable of doing a grid search to find the initial value. But this is not true: let us see a few R commands^{XXIII} to clear it up (that is, grid search is possible in R but we will go through another way):

```
R> attach(draper.data)
R> fit0 <- lm(log(Y)~t-1, draper)
R> fit0
Call:
lm(formula = log(Y) ~ t - 1, data = draper)

Coefficients:
          t
-0.2012XXIV

R> fit1 <- nls(Y~exp(-THETA*t), data=draper,
start=c(THETA=-0.2012))
R> fit1

Nonlinear regression model
  model: Y ~ exp(-THETA * t)
 data:  draper

      THETA
0.2034489

residual sum-of-squares:  0.0003018337

R> summary(fit1)

Formula: Y ~ exp(-THETA * t)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
THETA 0.203449   0.006002   33.90  0.00087 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01228 on 2 degrees of freedom
```

Alternatively, we could compute the sum of squares for all values of THETA = seq(0, .01, 100) in a loop, then find the minimum by eye. Also, for a problem that has several parameters, "expand.grid" will produce a grid, and we can compute the value of function and the sum of squares of residuals at every point in the grid in a single loop, etc.

^{XXIII} Thanks to Spencer Graves and Douglas Bates for the kind direction about this grid search using R.

^{XXIV} We would expect that (-0.2012) should provide a reasonable starting value for nls.

Using Statistical Software: NLREG nonlinear regression program

NLREG nonlinear regression program^{XXV} has a "Sweep" statement that can be used to try multiple starting points for a parameter or do a grid search if there are multiple parameters. However, it is rarely needed.

Here is an NLREG program to solve our example problem. The Sweep statement tries starting values for Theta from 1 to 10 in steps of 0.5. It turns out that the program doesn't need the Sweep statement because it quickly achieves convergence with the default starting value of 1.0. The optimum value of Theta to fit your function is 0.203448874; it was found in 10 iterations.

```
Variables t, Y;
Parameter Theta;
Function Y = EXP(-(THETA * t));
Sweep Theta = 0, 10, 0.5;
Plot;
Data;
1 0.80
4 0.45
16 0.04
```

And the out put is as follows :

```
NLREG version 6.1
Copyright (c) 1992-2004 Phillip H. SherrodXXVI.

Number of observations = 3
Maximum allowed number of iterations = 500
Convergence tolerance factor = 1.000000E-010
Stopped due to: Both parameter and relative function convergence.
Number of iterations performed = 8
Final sum of squared deviations = 3.0183370E-004
Final sum of deviations = -7.6585277E-003
Standard error of estimate = 0.0122848
Average deviation = 0.0080551
Maximum deviation for any observation = 0.0159119
Proportion of variance explained (R^2) = 0.9990 (99.90%)
Adjusted coefficient of multiple determination (Ra^2) = 0.9990 (99.90%)
Durbin-Watson test for autocorrelation = 1.810
Analysis completed 15-Apr-2004 05:51. Runtime = 0.44 seconds.
```

---- Descriptive Statistics for Variables ----

Variable	Minimum value	Maximum value	Mean value	Standard dev.
t	1	16	7	7.937254
Y	0.04	0.8	0.43	0.3803945

---- Calculated Parameter Values ----

Parameter	Initial guess	Final estimate	Standard error	t	Prob(t)
Theta	0.5	0.203448874	0.006002203	33.90	0.00087

^{XXV} Demonstration version (free for a month) is available at <http://www.nlreg.com>

^{XXVI} I am indebted to Phil Sherrod for writing the above example program for me personally.