

Required Statements

PROC NLIN statement -

This initiates the call to do nonlinear regression. Like other SAS PROCs, the **DATA=** option can be used to designate the required data set. Several other options are also available, most which allow control over the estimation process. The most important one for this discussion is **METHOD=**. Unlike linear regression, nonlinear regression is an iterative process. That is, several values for the model parameters are evaluated in an iterative fashion (algorithm) to find the best estimates. The way the iteration is done can vary and thus, the **METHOD=** option defines which iteration method to use. There is no "right" method and, in fact, several methods may be attempted before the estimation process converges satisfactorily. Some methods available are derivative based algorithms such as Gauss (default), Marquardt, and Gradient, as well as one that does not require derivatives, DUD. For most applications, however, the methods GAUSS or MARQUARDT would be adequate.

PARMS statement -

The PARMS statement tells SAS which parts of the model represent the parameters. This is not necessary in linear regression because each variable has one associated parameter (e.g. slopes or intercept). In nonlinear regression however, a variable may have one or more parameters associated with it. Thus, SAS must distinguish between parameters and variables. The PARMS statement also allows the user to set the initial values for each parameter before the iterative evaluation begins. These are called starting values and they play a very important role in the estimation

process. If the starting values are too far removed from the final estimates, the iterative process can become "lost" settling on a solution which is not correct (these are called local minima). It may be necessary to try several starting values in order to achieve a reasonable solution. One alternative is to give SAS a range of starting values in the PARMS statement. SAS will examine all possible combinations of the starting values and choose the set with the smallest SSE to initiate the iterative process. Caution should be used, however, since the evaluation process can be time consuming. An example of a simple PARMS statement might be:

parms a=1.25 b0=49.6 rho=.002;

Notice that several parameters can be listed, and that they are separated by a space. A more complex statement form indicating a range of starting values for say **a** and **b0** would be:

parms a=1 to 2 by .1 b0=40 to 60 by 1 rho=.002;

In this case, the range is separated by the word **to** and the search increment is specified with the word **by**. Note that SAS will have to evaluate $10 \times 20 \times 1 = 200$ combinations of parameters before deciding on starting values!

Because SAS must identify the parameters and their starting values, the PARM statement **must come before** the MODEL statement.

MODEL statement -

The MODEL statement defines the dependent variable, regressor variable(s), and their relationship to the parameters. Unlike MODEL statements in other PROCs,

there are no options available. An example for the MODEL statement using a form of the exponential function might be:

*model y = a*exp(b0*(x-rho));*

Here, **y** is the dependent variable, **x** is the regressor variable and **a**, **b0** and **rho** are the parameters to be estimated.

DER (derivative) statements -

Most of the iterative methods of estimation require a set of partial derivatives to be defined from the model. These derivatives are taken with respect to each parameter, thus there are as many DER statements as there are parameters. Each DER statement is identified by a naming convention that uses its associated parameter's name. Examples of DER statements for the model above would be:

der.a = *exp(b0*(x-rho));*
der.b0 = *a*(x-rho)*exp(b0*(x-rho));*
der.rho = *-a*b0*exp(b0*(x-rho));*

If the DER statements are left out, NLIN will use an iteration method which does not require derivatives called DUD (**D**on't **U**se **D**erivatives!). Note: this will happen no matter what is specified in the METHOD= option. Although this may sound appealing to the calculus-impaired, the use of this method precludes some types of output and diagnostics.

Optional Statements

BOUNDS statement -

It is sometimes necessary or desirable to limit the values that a parameter can legitimately take on. For example, in the model above, it may be known that the parameter **rho** must be between 0 and 1 or that **b0** should be positive. The syntax is similar to the simple PARMs statement, but parameters are separated by a comma:

bounds 0 < rho < 1, b0 >= 0;

It is not necessary to bound all (or any) parameters, but when they are, the initial starting values given in the PARMs statement **must** conform to the bounds.

OUTPUT statement -

The OUTPUT statement is used to transfer the information obtained with PROC NLIN out to other procedures. The most important information in this regard are the predicted function values and their associated residuals, which in turn provide valuable diagnostic features to assess the model fit. A good practice in nonlinear regression is to a) start with a plot of the data, b) determine a model form which is appropriate for the data, and c) evaluate after estimation with a plot of the predicted function against the observed values. This is a requirement because, unlike linear regression, nonlinear estimation may converge (find an estimate) at values which are inappropriate. In other words, the predicted function may be no where near the data. This is one of the dangers in nonlinear regression and the only way to tell if the fit is reasonable is to plot it. The following OUTPUT statement could be used:

```
output out=pred p=yhat student=resid l95m=lower u95m=upper;
```

This creates a data set called PRED which has in it, the predicted values (yhat), the studentized residuals (resid), and the lower and upper 95% confidence intervals (lower and upper). In addition, all the variables in the original data set are also included. This data set can then be used for subsequent plotting and statistical assessment.

PROGRAMMING statements -

PROC NLIN is different from every other SAS PROC in one feature. It will accept programming statements which are usually the domain of the DATA STEP. This exception allows great flexibility in the NLIN procedure. Examples of statements that can be used are the IF-THEN-ELSE clauses and assignment statements (e.g. $\text{ltime}=\log(\text{time})$). These may be needed if a more complex model form is to be fitted. An example might be:

```
proc nlin method=dud;  
  parms M=95 b=.02 L=5;  
  if x <= L then do;  
    model y=0;  
  end;  
  else do;  
    model y = M*(1 - exp(-b*(x - L)));  
  end;
```

This example sets up a segmented model which fits a flat line for x less than 5 and an exponential function for values greater than the value L. Note that the iteration method uses no derivatives.