PARAM

PARAM *parameter name* [*value* *parameter name* *value* ....] ;

Function:

PARAM is used to define parameters for the nonlinear estimation procedures and to assign starting values to them. To supply parameter starting values to PROBIT, TOBIT, SAMPSEL, and LOGIT, use the @START vector; see those procedures for further information.

Usage:

PARAM may be followed by as many argument pairs as desired (limited only by TSP's argument limit). Each pair is the name of the parameter followed by the value it is to be given. The parameter names may be that of new or previously defined variables. The value may be omitted, in which case the variable is given the value zero if it is new, or left unchanged if it has already been defined. Note: the keywords C and CONSTANT may not be used as parameters.

Procedures which estimate values for parameters defined by the PARAM statement are LSQ for nonlinear single and multi-equation least squares (including minimum distance estimators) and FIML. All other procedures treat parameters like constants (scalar variables) which have the arithmetic value they have been assigned, either by a PARAM statement or by later estimation. FORM(PARAM) can also be used to create parameters.

Example:

A common problem in nonlinear estimation is that one or more parameters may enter the model in a highly nonlinear fashion, making it difficult to estimate unless you have good starting values. In this example, we estimate a subset of the parameters of a model conditional on the value of another parameter, DELTA, and then reestimate with all the parameters free:

```
FRML INVEQ I = LAMBDA*I(-1) + ALPHA*GNP/(DELTA+R) ;
PARAM LAMBDA ALPHA ;
CONST DELTA 15 ;
LSQ INVEQ ;
PARAM DELTA ;
LSQ INVEQ ;
```

When the second LSQ is done, the starting values for LAMBDA and ALPHA will be those determined by the first estimation, while the starting value for DELTA will be 15, which it was assigned by the CONST statement.

Output:

PARAM produces no printed output; it stores the variables named in data storage with a type equal to parameter.