EQSUB

EQSUB (NAME=*new output equation name*, PRINT)
main equation name list of input equation names ;

Function:

EQSUB substitutes one or more equations into another. This is useful for estimation with several parameter restrictions, with long equations which have common terms, or for setting up a complicated model where the exogenous variables can be changed later by just changing one of the input equations.

Usage:

Define the main equation and the input equation(s) with FRML or IDENT statements. The EQSUB command substitutes each input equation into the main equation, in order from left to right. For each input equation, EQSUB looks for its dependent variable (or equation name, if there is no dependent variable) in the argument list of the main equation. If the dependent variable is found, the code from the input equation is inserted into the main equation, and the old variable name is deleted. For example:

```
FRML EQ1 Y = A + XB;
FRML EXB XB = X1*B1 + X2*B2;
EQSUB EQ1 EXB;
```

is equivalent to:

```
FRML EQ1 Y = A + X1*B1 + X2*B2;
```

The resulting equation replaces the main equation, unless the NAME= option is supplied. The DOT command is useful when there are several different main equations.

If you have many component input equations to define, it may be convenient to leave out the dependent variable name, so that you don't have to invent both a dependent variable and an equation name for each one. Such an equation is called "unnormalized" in TSP. For example,

```
FRML E Y1 - XB;  FRML XB B0 + B1*X1 + B2*X2;  ? change XB to add/delete exog. errors variables
FRML TOBIT LOGL = YPOS*( LNORM(E/SIGMA) - LOG(SIGMA) ) + YZERO*LCNORM(-XB/SIGMA);
EQSUB(NAME=TOBIT1) TOBIT E XB;
```

Note that both TOBIT and E depend on XB, so XB is substituted in last. There is no need to substitute XB into E separately. A separate substitution would still operate correctly, but it would result in larger and less efficient code. The new FRML TOBIT1 is created, and the original TOBIT is left untouched for later use.

On the other hand, normalized input equations are recommended for parameter restrictions. The same FRMLs can usually be used to both impose parameter restrictions, and to evaluate the restricted parameters after estimation with ANALYZ. If the input equation is normalized, ANALYZ can store the restricted parameter name. For example, in a translog model with symmetry imposed:

```
FRML EQ1 SH1 = A1 + B11*LP1 + B12*LP2 + B13*LP3;
FRML EQ2 SH2 = A2 + B12*LP1 + B22*LP2 + B23*LP3;
?Note: The last share equation is not used in estimation due to singularity.
?FRML EQ3 SH3 = A3 + B13*LP1 + B23*LP2 + B33*LP3;
```

EQSUB

```
FRML R13 B13 = -(B11+B12);          ? homogeneity/adding up constraints
FRML R23 B23 = -(B12+B22);
EQSUB EQ1 R13;  EQSUB EQ2 R23;
FRML LO1 A3 = 1 - (A1+A2);          ? left out params from final equation
FRML LO2 B33 = -(B12+B23);          ? note: this also depends on the restricted B23
EQSUB LO2 R23;
LSQ EQ1 EQ2;                        ? Estimate model with restrictions in place
ANALYZ LO1 LO2;                     ? Print and store values and standard errors for A3 and B33
```

EQSUB can also handle "lagged dependent variables" (as of 4/1996):

```
FRML U Y - (A + B*X + G*Z(-2));
FRML E U - RHO*U(-1);
EQSUB U E;
```

is equivalent to

```
FRML E Y - (A+B*X+G*Z(-2)) - RHO*(Y(-1) - (A+B*X(-1)+G*Z(-3)));
```

Note that when the EQSUB command is given, all the variables in the equations must exist (either as series, PARAMs, CONSTs, or other FRMLs), so that EQSUB will know which ones need to be lagged (the series and FRMLs), and which ones don't need lags (the PARAMs and CONSTs).

Options:

NAME= *new output equation name* supplies a new name for the output equation. If this option is not present, the main equation is overwritten by the new one.

PRINT/**NOPRINT** controls whether the output equation is printed.

Examples:

See above. See also the **TSP User's Guide** for many additional examples of using EQSUB to set up log likelihood equations for estimation by ML. Here is one more example, illustrating the use of DOT to substitute input equations F1 to F20 into main equations E1 to E8:

```
DOT E1-E8;
    EQSUB . F1-F20;
ENDDOT;
```

Output:

Normally, the output equation is stored silently, replacing the input equation, or creating a new equation. If the PRINT option is on, the output equation is printed.