IF

IF *scalar expression* ;

Function:

IF provides conditional execution of commands in TSP. It requires a subsequent THEN statement, and may also be followed by an ELSE statement if you wish to have a branch for a false result.

Usage:

The result of the scalar expression following IF is interpreted as true if it is larger than zero and false otherwise. If the result of the expression is a logical value itself (for example, TEST < 2.0) it will be given the value one for true and zero for false by TSP.

Only if the expression gives a true result will the statement following the next THEN ; statement be executed. If that statement is a DO statement, all the statements up to the closing ENDDO ; statement will be executed. If there is an ELSE ; statement subsequently, the statements following it will be executed in the same manner if the result of the IF was false.

Do not use the IF/THEN/ELSE commands to create variable transformations -- use the GENR command with logical expressions instead.

Examples:

See the examples for the ELSE statement.

Here is a simple example, where we print a one of two possible titles, depending on the results of a normality test on regression residuals:

```
OLSQ Y C X1 X2;
IF %JB < .05; THEN;   ? Note:  %JB is the P-value of the Jarque-Bera normality test
   TITLE 'Reject normality at 95% level';
ELSE;
   TITLE 'Cannot reject normality at 95% level';
```

Say you would like to create a 0/1 dummy variable, based on the value of INCOME larger than 40. In some packages, you would write commands like *IF INCOME>40 THEN D=1 ELSE D=0;* . However, in TSP, you can't use series like INCOME in IF statements, because they have more than one value (it creates complicated branching). Instead, use an implicit GENR command with the True (1) or False (0) result of an expression:

```
D = INCOME > 40;
```