MMAKE

MMAKE (VERT) *matrix name* *list of series* ;
or
MMAKE *vector name* *list of scalars* ;
or
MMAKE *(VERT)* *matrix name* *list of matrices* ;

Function:

MMAKE makes a new matrix by stacking a set of series or matrices, or a new vector from a set of scalars. In the series case, the new matrix normally has the number of rows equal to the number of observations and the number of columns equal to the number of series. In the matrix case, the new matrix has the number of rows equal to the common number of rows of the matrices and the number of columns equal to the total number of columns. The vector has the number of rows equal to the number of scalars.

MMAKE is the reverse of UNMAKE, which breaks a matrix into a set of series, or a vector into a set of scalars. To make a matrix from another matrix or vector by changing its type, dimensions, or transposing it, use the MFORM procedure.

Usage:

MMAKE's first argument is the name to be given to the new matrix, followed by a list of series or matrices which will form the columns of the new matrix. The number of series is limited only by the maximum size of the argument list (usually 2000 or more) and the space available in data storage for the new matrix.

Only the observations in the series which are within the current sample will be used, so you can select data for the matrix very conveniently by changing the SMPL. If you use matrices instead of series, the SMPL is ignored.

The matrix made by MMAKE will always be a general matrix and (when using series) its dimensions are normally NROW = @NOB (the number of observations) by NCOL = the number of input series. To change its type, use MFORM.

If the second and following arguments to MMAKE are scalars (CONSTs, PARAMs, or numbers), a vector will be created instead of a matrix. This is useful for creating vectors like @START or @COEF from estimated parameters or scalars created by UNMAKE from previous @COEF vectors.

Options:

VERT/**NOVERT** stacks the input series or matrices vertically instead of horizontally.

Examples:

If the current sample is SMPL 1 4 ; and there are two data series X1 = (1,2,3,4) and X2 = (9 8 7 6), the following command:

MMAKE

```
MMAKE X X1 X2 ;
```

creates the matrix

```
X =     1  9
        2  8
        3  7
        4  6
```

whereas

```
MMAKE(VERT) XV X1 X2;
```

creates the 10 x 1 vector XV:

```
1
2
3
4
9
8
7
6
```

Here is an example of adding a coefficient to a LOGIT estimation while retaining the starting values for the other coefficients from the previous estimation:

```
LOGIT(NCHOIC=2) WORK C SCHOOL EXPER RACE;
UNMAKE @COEF C1-C4;
MMAKE @START C1-C4 0;
LOGIT WORK C SCHOOL EXPER RACE MSTAT;
```

This example makes a matrix of regression output to be written to disk:

```
LOGIT WORK C SCHOOL EXPER RACE MSTAT ;
MMAKE REGTAB @COEF @SES @T ;
WRITE (FILE= "REGTAB.ASC", FORMAT= "(3F10.5)") REGTAB ;
```

The final example creates the partitioned matrix

```
A = D E F
    G H I
```

where D, E, and F are matrices with the same number of rows, D and G have the same number of columns, etc.

```
MMAKE DEF D E F;
MMAKE GHI G H I;
MMAKE(VERT) A DEF GHI;
```

Output:

MMAKE produces no printed output. A single matrix or vector is stored in data storage.