MFORM

MFORM (NROW=*# of rows in matrix*,NCOL=*# of columns in matrix*,
TRANS,TYPE=GENERAL or SYMMETRIC or TRIANG or DIAG)
variable name
or *new matrix* = *old variable*
or *new matrix* = *scalar*
or *old variable   new matrix* ;
or
MFORM(BAND,NROW=*nrows*) *new matrix = band_vector [corner]*;
or
MFORM(BLOCK) *new matrix = list of matrices*;

Function:

MFORM forms or reforms matrices used to make a matrix from a series or vector or to change the dimensions or type of an existing matrix. The matrix may be transposed as it is formed. Matrices can be renamed or copied without reformatting with the RENAME or COPY commands.

Usage:

If there is only one argument, either an existing matrix or series is transformed in place, or a new matrix is initialized to zero. The options specify the characteristics of the new matrix: the number of rows and columns, the type and whether it is to be transposed as it is formed.

When there is more than one argument, the old variable may be a series, a matrix, a vector, or a scalar. The new variable will be a matrix of the type specified on the command. If no type is specified, a general matrix is created. If the input variable is a series, it is retrieved under control of SMPL and only those observations in the current sample are placed in the matrix. If the input series or matrix is longer than NROW*NCOL, it is truncated (except in the case of the DIAG type - see the examples). If it is shorter, an error message is given, unless it is a scalar, in which case it is duplicated throughout the new matrix.

If you specify a type such as triangular which is not consistent with the input matrix, MFORM will change the input so that it is, i.e., the elements below the diagonal will be zeroed. You can use this feature, for example, to select the diagonal elements of a matrix and form a new matrix from them (like a **diag** function). UNMAKE can be used to form a series from the diagonal of a diagonal matrix.

Options:

BAND/**NOBAND** specifies that a (symmetric) band matrix is to be formed from a vector of band values, and optionally a symmetric corner matrix (for the upper left and lower right corners, to override the band values).

BLOCK/**NOBLOCK** specifies that a block diagonal matrix is to be formed from a list of symmetric or general matrices (by placing them along the diagonal, and putting zeros elsewhere). This is useful for composing VCOV matrices from several independent estimations for minimum distance estimation with LSQ (or for ANALYZ).

NROW= the number of rows in the matrix to be formed. This is required for a general matrix.

NCOL= the number of columns in the matrix to be formed. This is required for a general matrix.

Either NROW or NCOL must be specified for symmetric, triangular, or diagonal matrices, unless the input variable is a matrix and you wish the new matrix to have the same dimensions.

TRANS/**NOTRANS** specifies whether the input variable is to be transposed to produce a matrix of the type and dimensions specified. If the input variable is a matrix, the output matrix will have the number of rows equal to the number of columns of input and the number of columns equal to the number of rows of input. MFORM(TRANS) is the same as the MATRAN command.

TYPE=GENERAL or SYMMETRIC or TRIANG or DIAG specifies the type of the new matrix. GENERAL, the default, may be used for any rectangular or square matrix. SYMMETRIC implies that the matrix is equal to its transpose; only the lower triangle will be stored internally to save space. TRIANG implies that the matrix is upper triangular (has zeroes below the diagonal). DIAG means a matrix whose off-diagonal elements are zero. Only the diagonal is stored, and it is expanded before use. A series may be used as input to MFORM(DIAG).

Examples:

Suppose that we have a nine observation series called X containing the values 10,20,30,40,50,60,70,80,90. Each of the following MFORM commands will yield a different result:

```
SMPL 1,9;
MFORM (TYPE=GENERAL,NROW=3,NCOL=3) X ;

        yields X =  10   40   70
                    20   50   80
                    30   60   90
```

```
SMPL 1,6;
MFORM (TYPE=GENERAL,NROW=2,NCOL=3) X XMAT ;

        yields  XMAT =  10   30   50
                        20   40   60
```

```
MFORM(TRANS) XMATT=XMAT ;

        yields XMATT =   10   20
                         30   40
                         50   60
```

```
MFORM (TYPE=GENERAL,NROW=4,NCOL=3) X ;
```

gives an error message; the resulting matrix is too big for the amount of data available.

```
SMPL 1,9;
MFORM (TYPE=SYM,NROW=3) XSYM=X ;

        yields  XSYM =  10   20   30
                        20   50   60
                        30   60   90
```

```
MFORM (TYPE=TRIANG,NCOL=3) X ;
```

```
          yields  X =  10   40   70
                        0   50   80
                        0    0   90
```

```
MFORM (TYPE=DIAG,NCOL=3) X ;
```

```
          yields  X =  10    0    0
                        0   50    0
                        0    0   90
```

```
MFORM (TYPE=DIAG,NROW=9) X ;        ? or use MAT X = DIAG(X);
```

```
          yields X =  10    0    0    0    0    0    0    0    0
                       0   20    0    0    0    0    0    0    0
                       0    0   30    0    0    0    0    0    0
                       0    0    0   40    0    0    0    0    0
                       0    0    0    0   50    0    0    0    0
                       0    0    0    0    0   60    0    0    0
                       0    0    0    0    0    0   70    0    0
                       0    0    0    0    0    0    0   80    0
                       0    0    0    0    0    0    0    0   90
```

The next example shows how you can make a diagonal matrix from a column vector - if the input series or matrix is too short to make a diagonal matrix by selecting diagonal elements, the whole vector becomes the diagonal.

```
MFORM (TYPE=DIAG,NCOL=3) X = 2;    ? or  MAT X = 2*IDENT(3);
```

```
          yields  X =  2   0   0
                       0   2   0
                       0   0   2
```

A band matrix:

```
MMAKE BVEC 2 -1;
READ(NROW=2,TYPE=SYM) CORNER;   11 21 22;
MFORM(BAND,NROW=5) B5 = BVEC CORNER;
```

```
          yields B5 =  11   21    0    0    0
                       21   22   −1    0    0
                        0   −1    2   −1    0
                        0    0   −1   22   21
                        0    0    0   21   11
```

Output:

MFORM produces no printed output. A single matrix is stored in data storage.

MFORM