Guide to Matrix Routines in SPSS

By Ed Hasecke 4.99

Getting Started

Open syntax window in SPSS (File → New → Syntax). This is where we will be writing the matrix algebra routines that SPSS will run.

Two Beginning Points

1. The matrix program must begin with the statement
   MATRIX.
   and after all the formulas are entered, must end with the statement
   END MATRIX.
2. All statements must end with a period.

Defining Matrices

The first step in doing matrix routines is to define the matrices that we will be manipulating. In other words, we have to tell SPSS what our **X** and **Y** matrices are.

There are two ways to do this.

1. Create the matrix in the syntax.
   The command is:
   COMPUTE X={x_{11} , x_{12} ; x_{21} , x_{22}}.
   Each element in a row is separated by a comma and each row is separated by a semi-colon.
2. Pull variables out of existing data set.
   If your data is already in an SPSS data file (.sav), then you can pull variables out of the data set by name. First **open your data** in the Data Editor. Return to the syntax window and use the following command:
   GET X/VARIABLES=X0, X1.

NOTE: Don't forget a column vector of 1s. You will either have to enter a 1 at the beginning of each column or create a variable in your data set.

Program Commands

Every line of your program will begin with one of three commands.

1. COMPUTE
   This command creates new matrices. It is the command that says, "Hey, I'm going to write an equation!"
   The basic form is:
   COMPUTE NEWMATRIX = f(OLDMATRIX).
   i.e. COMPUTE NEWX = X + X.
   We will return to the possible functions (f( )) later.

2. PRINT

This command tells SPSS to print the matrix to your output.

The basic form is:

PRINT MATRIX.

i.e. PRINT X. (to print the X matrix)

3. COMMENT

This tells the program to ignore whatever you write after it. This is not necessary for making a successgul program, but I suggest that you use it to annotate your program. It helps when you return to your program each time, you can tell what you are trying to do at each stage of your syntax. It also helps Prof. Pollins in grading.

The basic form is:

COMMENT WHATEVER YOU WANT.

i.e. COMMENT FIND PREDICTED Y.

Operations (Functions)

The Usual Suspects

+ = addition
- = subtraction
* = multiplication
/ = division

element-by-element operations on two vectors of identical dimensions: &* = element-by-element multiplication. &/ = element-by-element division

The Matrix Operations

T( ) = transpose of whatever is in parentheses

i.e. T(X) = X transpose = X'

INV( ) = inverse of whatever is in parentheses

i.e. INV(X) = X inverse = X^{-1}

Other Helpful (Necessary) Operations

NROW( ) = number of rows in the matrix in parentheses

i.e. NROW(X) = number of rows in X

NCOL( ) = number of columns in the matrix in parentheses

i.e. NCOL(X) = number of columns in X

DIAG( ) = takes the main diagonal of the matrix in parentheses and makes a column vector.

i.e. DIAG(X) = main diagonal of X.

SQRT( ) = square root of the value in parentheses.

i.e. SQRT(4) = 2.

CSUM( ) = sum of all values in the column vector in parentheses.

i.e. CSUM(Y) = sum of all elements in Y.

Running the Program

When you are done with your code, highlight the code and press the RUN key on the toolbar. Or, go to the RUN menu and choose SELECTION.

Example

Suppose I had the following 3x3 matrix.

2	7	4
3	1	5
6	4	3

The following is the routine necessary to define the matrix, transpose it, find the inverse of the original matrix, and finally check the inverse to make sure it is right.

```
MATRIX.
COMMENT DEFINE MATRIX.
COMPUTE X={2,7,4;3,1,5;6,4,3}.

COMMENT FIND X TRANSPOSE.
COMPUTE TRANS=T(X).
PRINT TRANS.

COMMENT FIND X INVERSE.
COMPUTE INVERSE=INV(X).
PRINT INVERSE.

COMMENT CHECK INVERSE.
COMPUTE CHECK=X*INVERSE.
PRINT CHECK.
END MATRIX.
```

Notice that the syntax begins with MATRIX and ends with END MATRIX. Also note that there is a period at the end of each statement. Below is the output from this routine. Notice that X does not appear because I did not include the PRINT X. command. This was done to illustrate that not all output must be shown. When you do the homework, only print out the necessary values. For example, it is a waste of paper to print out the X and Y matrices (especially when the dimensions are large).

```
Run MATRIX procedure:

TRANS
  2  3  6
  7  1  4
  4  5  3

INVERSE
  -.1240875912  -.0364963504   .2262773723
   .1532846715  -.1313868613   .0145985401
   .0437956204   .2481751825  -.1386861314

CHECK
   1.000000000   .000000000   .000000000
    .000000000  1.000000000   .000000000
    .000000000   .000000000  1.000000000

------ END MATRIX -----
```