Program Development (SAS IML)

$\tau\rho$

Review SAS IML

Functions for generating matrix

- BLOCK Function
- I Function
- J Function
- REPEAT Function
- SHAPE Function

BLOCK

- **Forms block-diagonal matrices**
- The BLOCK function creates a new block-diagonal matrix from all the matrices specified in the argument matrices. Up to 15 matrices can be specified. The matrices are combined diagonally to form a new matrix.
- For example, the statement

```
block(a,b,c);
```

produces a matrix of the form

$$\begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{bmatrix}$$

Example 1

```
proc iml;
reset print;
a={1 2 3, 6 5 4};
b={2 2 2, 4 4 4};
c=block(a,b);

d={3 3,7 7};
e=block(a,d);
```

```
a1={1 2};
a2={3 4};
a3={5 6};
a4={7 8};
a5=10;
aa=block(block(a1,a2),block(a3,a4),a5);
```

I

- **Creates an identity matrix**
- The I function creates an identity matrix with *dimension* rows and columns. The diagonal elements of an identity matrix are 1s; all other elements are 0s. The value of *dimension* must be an integer greater than or equal to 1. Noninteger operands are truncated to their integer part.
- For example, the statement

```
A=I(3);
```

yields the result

$$A=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

Example 2

```
proc iml;
reset print;
a=I(5);
b=block(I(3),I(4));

c=block(block(I(1),I(2),I(3)),I(2));
```

J

- **Creates a matrix of identical values**

J(*nrow<, ncol<, value>>***)**

- The inputs to the J function are as follows:
  - *nrow* is a numeric matrix or literal giving the number of rows.
  - *ncol* is a numeric matrix or literal giving the number of columns.
  - *value* is a numeric or character matrix or literal for filling the rows and columns of the matrix.
- The J function creates a matrix with *nrow* rows and *ncol* columns with all elements equal to *value*. If *ncol* is not specified, it defaults to *nrow*. If *value* is not specified, it defaults to 1.

Example 3

```
proc iml;
reset print;
a=J(5);
b=block(I(2),J(3),I(2));

c=J(5,2,'xyz');
d=J(9,9);
```

REPEAT

- **Creates a new matrix of repeated values**

  `REPEAT(` *`matrix, nrow, ncol`***`)`**

- The inputs to the REPEAT function are as follows:
  - *matrix* is a numeric matrix or literal.
  - *nrow* gives the number of times *matrix* is repeated across rows.
  - *ncol* gives the number of times *matrix* is repeated across columns.
- The REPEAT function creates a new matrix by repeating the values of the argument matrix *nrow***ncol* times, *ncol* times across the rows, and *nrow* times down the columns. The *matrix* argument can be numeric or character.

Example 4

```
proc iml;
reset print;
x={1 2,
   3 4};
y=repeat(x,2,3);


b1=block(x,I(1));
bb=repeat(b1,1,2);


cc=repeat(block(J(1,3),I(3),x),2,1);

dd=block(repeat(I(2),2,2),repeat(J(4,1,-1),2,2));
```

SHAPE

- **Reshapes and repeats values**

SHAPE(*matrix<, nrow<, ncol<, pad-value>>>***)**

- The inputs to the SHAPE function are as follows:
  - *matrix* is a numeric or character matrix or literal.
  - *nrow* gives the number of rows of the new matrix.
  - *ncol* gives the number of columns of the new matrix.
  - *pad-value* is a fill value.

SHAPE (2)

- The SHAPE function shapes a new matrix from a matrix with different dimensions; *nrow* specifies the number of rows, and *ncol* specifies the number of columns in the new matrix. The operator works for both numeric and character operands. The three ways of using the function are outlined below:
  - If only *nrow* is specified, the number of columns is determined as the number of elements in the object matrix divided by *nrow*. The number of elements must be exactly divisible; otherwise, a conformability error is diagnosed.
  - If both *nrow* and *ncol* are specified, but not *pad-value*, the result is obtained moving along the rows until the desired number of elements is obtained. The operation cycles back to the beginning of the object matrix to get more elements, if needed.
  - If *pad-value* is specified, the operation moves the elements of the object matrix first and then fills in any extra positions in the result with the *pad-value*.
  - If *nrow* or *ncol* is specified as 0, the number of rows or columns, respectively, becomes the number of values divided by *ncol* or *nrow*.

Example 5

```
proc iml;
reset print;
r1=shape(12,3,4);
r2=shape(77,1,5);

s={1 2, 3 4, 5 6};
r3=shape(s,2);

r4=shape({99 31},3,3);

r5=shape({1 2,3 4},5,2);

r6=shape(s,3,3,7);
```

Exercises

1. Create a simple program to build the matrix in the experiment design below:

a.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 2 & 1 \\ 1 & 1 & 0 & 0 & 2 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 & 2 & 2 & 2 \end{bmatrix}$$

b.

	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
ROW1	1	1	0	1	0	0	0	1	2	3
ROW2	1	1	0	1	0	0	0	4	7	7
ROW3	1	1	0	0	1	0	0	1	2	3
ROW4	1	1	0	0	1	0	0	4	7	7
ROW5	1	0	1	0	0	1	0	1	2	3
ROW6	1	0	1	0	0	1	0	4	7	7
ROW7	1	0	1	0	0	0	1	1	2	3
ROW8	1	0	1	0	0	0	1	4	7	7

c.

	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
ROW1	1	1	0	1	0	0	0	1	2	3
ROW2	1	1	0	1	0	0	0	4	7	7
ROW3	1	1	0	0	1	0	0	1	2	3
ROW4	1	1	0	0	1	0	0	4	7	7
ROW5	1	0	1	0	0	1	0	1	2	3
ROW6	1	0	1	0	0	1	0	4	7	7
ROW7	1	0	1	0	0	0	1	1	2	3
ROW8	1	0	1	0	0	0	1	4	7	7

Exercises (2)

2. Specify the ordo and the output of programs below:
   a. `d=block(repeat(I(2),2,2),block(J(2,2),J(2,2)));`
   b. `e=shape(block(1,2,3),4,5,7);`
   c. `f=repeat(block(shape({1 2},2,3),J(1)),1,2);`

3. Count the results of matrix operation below
   a. `g=shape({1 2},3,3)*J(3);`
   b. `f=inv(block(1,2,3,4,5))*(block(3,6,9,12,15));`
   c. `h=shape({1 2 3},6,2)`*J(6,1);`

Answer

```
proc iml;
reset print;
a=block(J(2,2),1,J(3,1))||(shape({1 2},3,3)//J(3,3,2));
```

```
b=repeat(shape({1 2,3 4},2,3,7),4,1);
c=J(8,1)||block(J(4,1),J(4,1))||block(J(2,1),J(2,1),J(2
,1),J(2,1));
hasil=c||b;
```

Answer (2)

```
G      3 rows      3 cols    (numeric)

             4         4         4
             5         5         5
             4         4         4

F      5 rows      5 cols    (numeric)

     3         0         0         0         0
     0         3         0         0         0
     0         0         3         0         0
     0         0         0         3         0
     0         0         0         0         3

H      2 rows      1 col     (numeric)

                    12
                    12
```

Thank you ☺