SAS notes for IML – Part 2

Reading a data from a matrix to a dataset:

Suppose that the dataset was entered in a matrix **X**:20x4 in SAS/IML. To read this data into a dataset in SAS/BASE called 'computer' with the variables named z1, z2, y1 and y2 the following steps can be entered in SAS/BASE.

```
create computer from x [colname = {z1 z2 y1 y2} ];
append from x;
```

Reading data from a dataset to a matrix:

Suppose the dataset called 'computer' is in the permanent library of SAS called wst311 and has four variables named z1, z2, y1 and y2. The syntax in SAS/IML to read this dataset into a matrix called **X** is as follows.

```
proc iml; reset nolog;
use wst311.computer;
read all var{z1 z2 y1 y2} into x;
```

Writing a program to be general: example

Try to write the SAS program in such a way that you can do calculations on data matrices by only changing the data matrix and not other values, for example the sample size n.

Consider the following dataset with four observations and two variables V1 and V2.

Obs	V1	V2
1	1	56
2	2	34
3	3	12
4	6	8

In SAS/IML the sample mean, sample covariance and sample correlation matrices can be determined as follows.

```
proc iml; reset nolog;
x={1 56, 2 34, 3 12, 6 8};
n=nrow(x);                  * this gives the sample size;
j1=j(n,1,1);                * this gives the nx1 column vector that can be used to calculate the sample mean
                              and covariance matrix;
xbar=1/n # x` * j1;         * sample mean;
s=(1/(n-1))#x`*(I(n)-1/n#j1*j1`)*x;
                            * sample covariance matrix, I(n) is nxn identity matrix;
d=inv(sqrt(diag(s)));       * diag(s) gives a nxn matrix with elements equal to those of the diagonal of s and all
                              other elements equal to 0, sqrt calculates square root of each element of the
                              matrix, inv determines inverse of the matrix;
r=d*s*d;                    * sample correlation matrix;
print xbar s r;             * print sample mean, covariance and correlation matrices;
```

Using SAS/IML to calculate eigenvalues and eigenvectors

The following program will give the eigenvalues (lambda) and the corresponding normalized eigenvectors for the square matrix s in the previous example.

```
proc iml; reset nolog;
call eigen(lambda,h,s);
print lambda h;
```

Calculating the sum or product of the elements of a vector

```
prodx=x[#];                 * gives the product of the elements of the vector x;
sumx=x[+];                  * gives the sum of the elements of the vector x;
```

Selecting specific elements in a matrix/vector

Consider a 3x3 matrix entered into SAS/IML. To select the elements in the last two rows and the first two columns of **X**, the following syntax can be used.

```
proc iml; reset nolog;
x={1 3 5, 7 9 11, 13 15 17};
xa=x[2:3,1:2];
```

Combining matrices and vectors to create a new matrix

Consider the matrix **X**:2x2, the row vector **rv**:1x2 and the column vector **cv**:3x1. These can be combined to form a new matrix **Y**:3x3.

```
proc iml; reset nolog;
x={1 2, 4 5};
rv={7 8};
cv={3, 6, 9};
y=(x//rv)||cv;           * the symbol // combines the rv below x, || combines the cv to the right of (x//rv);
```

Summing the elements of a row or column vector of a matrix

Consider a matrix **X**:3x4. The following SAS/IML program calculates the sum of the row elements and the sum of the column elements. Two methods are given.

```
proc iml; reset nolog;
x={1 1 1 1, 2 2 2 2, 3 3 3 3};

/* method 1*/
sumrow=x[,+];            * adds the row elements of the matrix x;
sumcol=x[+,];            * adds the column elements of the matrix x;

/* method 2*/
nrow=nrow(x);            * determines the number of rows in the matrix x;
ncol=ncol(x);            * determines the number of columns in the matrix x;
j1=j(nrow,1,1);          * create a column vector j1 with nrow 1's as elements, in this case 3;
j2=j(ncol,1,1);          * create a column vector j2 with ncol 1's as elements, in this case 4;
sumrow2=x*j2;            * create a column vector sumrow2 with elements the sum of the rows of x;
sumcol2=j1`*x;           * create a row vector sumcol2 with elements the sum of the columns of x;
```

Generating random vectors from a multivariate normal distribution with mean 0 and covariance matrix the identity matrix

Suppose **Z**:3x1 is a random vector where the elements are all independent N(0,1). Generate a random sample of size 10 of these random vectors. Let **Y**:3x10 be the matrix containing these 10 3x1 vectors. Two methods to enter this in SAS/IML are given.

```
proc iml; reset nolog;

/* method 1*/
seed=j(3,10,0);
y=normal(seed);

/* method 2*/
y2=rannor(j(3,10,0));
```