WST312 Stochastic Processes 2015

Introduction to PROC IML

Aim: Getting acquainted with SAS and VBA. Do the questions for yourself. These questions are not for marks/hand-in.

SAS Proc IML

```
proc iml; /* General Structure */
/* optional */
options nocenter pagesize = 15 nodate;
reset nolog;
/* code ....
.....
.....
.....
.....
*/

quit;
```

print : use to output which data you would like to see once the program is run e.g. a matrix or variable created
nocenter : keep output on left of the screen (otherwise use **center** (default))
pagesize = n : allows you to decide on the number of lines of output per page
nodate : allows the option of no date and time in output
reset nolog : specifies the output to the output screen (default)

1. Entering a Matrix

```
proc iml; /* Entering a Matrix */
reset nolog;
P = {1 2,4 4};
print P;
quit;
```

Note that a matrix is entered with { } brackets.
(i) Type the above code for yourself and describe (**and** give print outs of) the output i.e. what matrix is produced?

(i) Does the comma separate the rows or columns?

2. Matrix Operations

*	: matrix multiplication
+	: addition
-	: subtraction
#	: element-wise multiplication
**	: matrix power
##	: element-wise power
@	: direct product
`	: transpose (note that this is the key on the left of the keyboard with the ~)
[,]	: selects submatrices e.g. [2,3:4] selects row 2 and columns 3 to 4
||	: concatenates matrices horizontally
//	: concatenates matrices vertically
[+,]	: adds the columns i.e. returns a row vector
[,+]	: adds the rows i.e. returns a column vector
[+]	: adds all the elements
J(m,n,c)	: creates an $m \times n$ matrix with all elements equal to c (important for initializing matrices)
max(M)	: returns the largest element in a matrix M
min(M)	: returns the smallest element in a matrix M
inv(M)	: returns the inverse of a matrix M
I(n)	: creates an $n \times n$ identity matrix

EXAMPLE

```
proc iml; /* Example */
reset nolog;
M = {1 3, 2 4};
N = {5 7, 6 8};
MN = M*N;
print MN;
quit;
```

Consider the following matrices: $A = \begin{bmatrix} 3 & 2 & 0.5 \\ 5 & 3 & 9 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 2 & 5 \\ 5 & 6 \end{bmatrix}, \quad D = \begin{bmatrix} 0.1 & 0.1 \\ 5 & 5 \end{bmatrix}$. For all the following questions type your own code to determine the required calculations using ***proc iml*** and include the code and output when you hand this assignment in.

(i) CD, DA

(ii) C^2, D^{10}

(iii) $C \# D, C @ D$

(iv) $A\#\#2$

(v) B`

(vi) $A[1,1], A[1,3], A[1,1] \,||\, A[1,3], A[1,1] \,//\, A[1,2]$

(vii) $B[+,], B[+], A[+,], D[,+]$

(viii) J(2,2,1), J(2,2,0)

(ix) I(2), {1,0} // {0,1}

(x) $inv(C), inv(D)$

(xi) max(A), min(A)

3. Logic Statements

- if *expression* then *statement*;
- if *expression* then *statement*
  else *statement*;
- if *expression* then
  do;
  statement;
  end;
- do i = 1 to n;
  ⋮
  end;

NOTE where the semi-colons are. Also see table A1.1 in the SAS notes.
Consider the following sample code:

```
proc iml; /* Logic */
reset nolog;
x = 0;
y = 3;
count = 0;
answer = 9;
if x=0 then print 'x=0';

if x = 0 & y = 3 then count=count+1;
print count;

if x = 0 then
do;
      if y ^= 0 then print 'x = 0, y not 0';
      else print 'x = 0, y = 0';
end;
else print 'x not 0';

if answer = 9 then
do;
      count = count + 1;
      print 'answer = 9';
end;
print count;

do i = 1 to 20;
      x = x + 1;
end;
print x;

quit;
```

Type your own NEW code to help you understand the logic statements above by coming up with similar (but UNIQUE) examples to the sample code above. The more initiative taken, the more marks that will be awarded. Include the corresponding output with the code.

4. **A Stochastic Process**

Consider the matrix

$$P = \begin{bmatrix} 0.11 & 0.89 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.4 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.6 & 0.15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0.4 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.4 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.2 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.3 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.65 & 0.25 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.75 & 0.05 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.12 & 0.88 \end{bmatrix}.$$

(i) Write your own code to which checks whether or not this matrix is a stochastic matrix i.e. that every element is a valid probability and the elements in each row add to 1. Your code should print out a comment saying 'yes, it is a stochastic matrix' or 'no, it isn't a stochastic matrix).

(ii) We are often interested in $\lim_{n\to\infty} P^n$. For this specific matrix this limit will be of the form

$$\Pi = \begin{bmatrix} \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \\ \overline{\pi} \end{bmatrix},$$

where each $\overline{\pi}$ is an identical row vector

$$\overline{\pi} = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 & \pi_5 & \pi_6 & \pi_7 & \pi_8 & \pi_9 & \pi_{10} \end{bmatrix},$$

such that $\sum_{i=1}^{10} \pi_i = 1$. By calculating P^n for various increasing values of n, determine the values of $\pi_1, \pi_2, ..., \pi_{10}$ accurate to 4 decimal places.

π_1		π_6	
π_2		π_7	
π_3		π_8	
π_4		π_9	
π_5		π_{10}	

(Remember to include all code and output for the above questions.)