STK 210: Practical 3

1 PROC IML: Combinatorial Functions and SAS Data Sets

1.1 Combinatorial Functions

1.1.1 RANPERM Function

The RANPERM function generates random permutations taken from a set of n elements.

Syntax:

```
RANPERM(set<,numperm>);
```

If `set` is a vector, the RANPERM function returns a **permutation** of the `set`. By default, the RANPERM function returns a single random combination with one row and n columns. If the `numperm` argument is specified, the function returns a matrix with `numperm` rows and n columns.

SAS Program:

```
proc iml;
call randseed(135,1);
a = ranperm({C B A}, 5);
print a;
```

Output:

```
  a
B A C
C A B
C B A
C A B
B A C
```

NB: The RANDSEED subroutine creates an initial random seed for subsequent random generated calls.

Syntax:

```
CALL RANDSEED(seed<,reinit>);
```

If RANDSEED is not called, an initial seed is generated from the system clock. This subroutine is normally used when it is desirable to reproduce the same random number stream in different PROC IML sessions. Otherwise you will get different random generated values every time. The optional `reinit` parameter controls whether the seed is reinitialized within the same PROC IML session. If it is set to one, identical seeds produce the same random number sequence; otherwise a second call to RANDSEED within the same PROC IML session is ignored.

1.1.2 RANCOMB Function

The RANCOMB function generates random combinations of k elements taken from a set of n elements.

Syntax:

```
RANCOMB(set,k<,numcomb>);
```

If `set` is a vector, the RANCOMB function returns a **combination** k elements of the `set`. By default, the RANCOMB function returns a single random combination with one row and k columns. If the `numcomb` argument is specified, the function returns a matrix with `numcomb` rows and k columns. Because we are working with combinations and the arrangement does not play a role you will note that the generated combinations are all sorted.

SAS Program:

```
call randseed(246,1);
b = rancomb({A B C D}, 3, 7);
print b;
```

Output:

```
  b
A C D
A B C
A C D
A B C
A C D
B C D
B C D
```

1.1.3 RANPERK Function

The RANPERK function generates a random permutation of k elements from a set of n elements.

Syntax:

```
RANPERK(set,k<,numperm>);
```

If `set` is a vector, the RANPERK function returns a **permutation** of k elements of the `set`. By default, the RANPERK function returns a single random permutation with one row and k columns. If the `numperm` argument is specified, the function returns a matrix with `numperm` rows and k columns. Each row of the returned matrix represents a single random draw that is not sorted.

SAS Program:

```
call randseed(123,1);
c = ranperk({A B C D E}, 3, 7);
print c;
```

Output:

```
  c
C B A
B C D
B A C
A E C
C E B
E C A
A C E
```

Caution: It seems to me that this function is not working properly when permutations of size k or $(k-1)$ is to be selected ?!?!?

1.1.4 FACT function

Computes a factorial.

Syntax:

`FACT(expression)`

where `expression` specifies any valid expression that evaluates to a numeric value. The mathematical representation of the FACT function is given by the following equation:

$$\texttt{FACT(n)}= n!$$

with $n \geq 0$.

Example

Statement	Result
`x=fact(5);`	120

1.1.5 PERM function

Computes the number of permutations of n items that are taken r at a time.

Syntax:

`PERM(n[,r])`

If `r` is omitted, the function returns the factorial of `n`. The mathematical representation of the PERM function is given by the following equation

$$\texttt{PERM(n,r)}= \frac{n!}{(n-r)!}$$

with $n \geq 0$, $r \geq 0$ and $n \geq r$.

Example

Statement	Result
`y=perm(5, 2);`	20

1.1.6 COMB function

Computes the number of combinations of n elements taken r at a time.

Syntax:

```
COMB(n,r)
```

The mathematical representation of the COMB function is given by the following equation

$$\texttt{COMB(n,r)} = \frac{n!}{r!\,(n-r)!}$$

with $n \geq 0$, $r \geq 0$ and $n \geq r$.

Example

Statement	Result
`z=comb(20, 4);`	4845

1.2 Working with SAS datasets

1.2.1 Creating SAS Data Set from a Matrix

SAS/IML software provides the capability to create a new SAS data set from a matrix. You can use the CREATE and APPEND statements to create a SAS data set from a matrix. An $n \times p$ matrix creates a SAS data set with p variables and n observations. That is, the columns of the matrix become the data set variables, and the rows of the matrix become the observations. The CREATE statement creates the new SAS data set, and the APPEND statement writes the observations.

Syntax:

```
CREATE SAS-data-set FROM matrix-name <[COLNAME=column-name]> ;
APPEND <FROM matrix>;
```

SAS Program:

```
proc iml;
weight={
189 165,
145 124,
210 192,
194 177,
127 118};
matrix_kg=round(0.453592*weight); * Scalar multiplication;
create data_kg from matrix_kg[colname={'start' 'end'}];
append from matrix_kg;

proc print data=data_kg;
title 'Health Club Data';
run;
```

SAS Output:

```
Health Club Data

Obs    start    end
 1       86      75
 2       66      56
 3       95      87
 4       88      80
 5       58      54
```

The variables in the SAS data set are called START and END. If you do not specify the COLNAME= option, the variables in the new data set are named COL1, COL2, and so forth.

1.2.2 Creating a Matrix from a SAS Data Set

Use the READ statement to read variables or records from the current SAS data set into column matrices of the VAR clause or into the single matrix of the INTO clause. When the INTO clause is used, each variable in the VAR clause becomes a column of the target matrix, and all variables in the VAR clause must be of the same type. If you specify no VAR clause, the default variables for the INTO clause are all numeric variables. Read all character variables into a target matrix by using VAR _CHAR_.

Syntax:'

```
USE SAS-data-set;
READ all <VAR operand> <WHERE(expression)> <INTO name <[COLNAME=column-name]>> ;
```

SAS Program:

```
data health; set data_kg;
loss=start-end;
run;

proc iml;
use health;
read all var{start end loss} into a;
print a;
quit;
```

SAS Output:

```
              a
       86        75        11
       66        56        10
       95        87         8
       88        80         8
       58        54         4
```

2 Exercise

1. **RSA Lotto:**

   - A participant of the LOTTO/LOTTO Plus Game has to choose 6 numbers from 1 to 49 (without replacement).
   - The winning numbers is always sorted from the smallest to the largest i.e. the ordening does not play a role.
   - John decides to use the RANCOMB function in SAS to help him to choose six numbers per ticket for the lottery.

   Note: You can create the row vector $\mathbf{x} = \begin{pmatrix} 1 & 2 & 3 & \cdots & 49 \end{pmatrix}$ by making use of the statement

   `x=1:49;`

   (a) Use the RANDSEED call with a seed of `711` to select 6 numbers for a **single** LOTTO ticket. $3; 6; 11; 24; 34; 37$

   (b) Use the RANDSEED call with a seed of `219` to select 6 numbers for **ten** LOTTO tickets.

   i. Print the numbers of the 10 LOTTO tickets.

   ii. Print the value of the 2nd number of the 3rd LOTTO ticket. 5

   iii. Print the value of the 6th number of the 4th LOTTO ticket. 49

   iv. Print the values of the 5th LOTTO ticket. $3; 9; 21; 22; 23; 34$

   v. Print the values of the 7th LOTTO ticket. $6; 12; 14; 30; 41; 46$

   vi. Print the smallest numbers of the 10 LOTTO tickets.
   Note: The smallest numbers are always listed first.

   vii. Print the largest numbers of the 10 LOTTO tickets.
   Note: The largest numbers are always listed last.

   (c) Use the RANDSEED call with a seed of `813` to select 6 numbers for 50 LOTTO tickets.

   i. Print the values of the 10th ticket. $2; 22; 39; 40; 42; 45$

   ii. Print the minimum value of the 20th ticket. 2

   iii. Print the maximum value of the 30th ticket. 44

   (d) Calculate the total number of possible LOTTO tickets. 13983816

2. **The Basic Counting Rule:**

   (a) How many different "routes" can a parcel follow if we have:

   - 2 collection points
   - 3 vehicles
   - 4 delivery points

   i. Calculate the total number by making use of PROC IML. 24

   ii. Use a DO loop to list all the experimental outcomes. See SAS Program `Q2.SAS`.

   (b) In how many different ways can one answer all the questions of a true-false test consisting of 20 questions? 1048576

3. **Permutations:**

(a) How many different arrangements can be formed of the letters in the word NUMBERS.

(b) Consider the word CARD.

i. Calculate how many permutations can be formed of the letters. 24

ii. Use the RANDSEED call with a seed of 425 to select a single permutation of the letters. $RDCA$

iii. Use the RANDSEED call with a seed of 513 to select 10 permutations of the letters. Print the first 5 permutations.

iv. Use the RANDSEED call with a seed of 876 to select 1000 permutations of the letters. Use the matrix $\mathbf{X}$ to store the 1000 permutations.

A. Print the 100$^{\text{th}}$ permutation of the 1000 generated pemutations. $DRAC$

B. Concatenate the 4 columns of the matrix $\mathbf{X}$ with the statement

```
r=concat(x[,1],x[,2],x[,3],x[,4]);
```

to create a single column vector $\mathbf{r}$.

C. Create the SAS data set DDD with the variable LETTERS containing the values of the column vector $\mathbf{r}$.

D. Use PROC FREQ to list the frequency distribution of the different permutations.

- Count the total number of different arrangements / permutations. 24
- How many times did the permutation DCRA occur? 46
- What percentage of the times did the permutation CADR occur? 5

(c) Consider the word BOOK.

i. Calculate how many permutations can be formed of the letters. 12

ii. Use the RANDSEED call with a seed of 912 to select 1000 permutations of the letters. Create a SAS data set with a variable with all the permutations of the word BOOK. Use PROC FREQ to list all the different arrangements.

A. Count the total number of different arrangements / permutations. 12

B. How many times did the permutation KBOO occur? 86

C. What percentage of the times did the permutation OOKB occur? 8.8

4. The Statistics and Economics marks for $n = 45$ students are given in the SAS data set MARKS in the SAS program Q4.SAS.

	Statistics	Economics
i	x	y
1	58	66
2	60	67
3	52	55
4	57	68
5	57	74
⋮	⋮	⋮
43	90	84
44	35	42
45	65	67

(a) Use PROC UNIVARIATE to give the value of the following for the Statistics marks (x):

i. $\sum_{i=1}^{n} x_i$ 2727

ii. $\frac{1}{n}\sum_{i=1}^{n} x_i$ 60.6

iii. $\sum_{i=1}^{n} x_i^2$ 171147

iv. $\sum_{i=1}^{n} (x_i - \bar{x})^2$ 5890.8

v. $\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2$ 133.88

vi. $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2}$ Interpret this value. 11.57073

- Average deviation of values from $\bar{x} = 60.6$ is 11.57073.

(b) Use PROC CORR to calculate the value of the following:

i. $s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2$ 133.88

ii. $s_y^2 = \frac{1}{n-1}\sum_{i=1}^{n} (y_i - \bar{y})^2$ 81.0636

iii. $s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$ 85.9727

iv. $r = \dfrac{s_{xy}}{s_x s_y}$ Interpret this value. 0.82525

- Strong positive linear relationship.

(c) Use PROC IML to create the matrix $\mathbf{M}$ where the first column is the Statistics marks and the second column is the Economics marks. Print the first 5 rows of the matrix M.

i. Create the following two column vectors:

- $\mathbf{x}$ containing all the Statistics marks
- $\mathbf{y}$ containing all the Economics marks

and calculate the following values in PROC IML:

A.	$\sum_{i=1}^{n} x_i$	2727
B.	$\sum_{i=1}^{n} x_i^2$	171147
C.	$\sum_{i=10}^{20} x_i$	658
D.	$\bar{x}$	60.6
E.	$\sum_{i=1}^{n} (x_i - \bar{x})^2$	5890.8
F.	$\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$	133.88182
G.	$\sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$	11.570731
H.	$\sum_{i=1}^{n} y_i$	3012
I.	$\sum_{i=1}^{n} y_i^2$	205170
J.	$\sum_{i=20}^{30} y_i$	731
K.	$\bar{y}$	66.933
L.	$\sum_{i=1}^{n} (y_i - \bar{y})^2$	3566.8
M.	$\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2$	81.0636
N.	$\sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2}$	9.00353
O.	$\sum_{i=1}^{n} x_i y_i$	186310
P.	$\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$	85.972727
Q.	$r_{xy} = \frac{s_{xy}}{s_x s_y} = 0.82525$	0.8252525

ii. Use the following statistical functions and compare your results with the previous questions.

- `MEAN(M)` `VAR(M)` `STD(M)`
- `COV(M)` `CORR(M)`

SAS Output: Statistical Functions Question 4(c)ii

```
 mean                 var                 std
     60.6 66.933333 133.88182 81.063636 11.570731 9.0035347

      cov                 corr
133.88182 85.972727         1 0.8252525
85.972727 81.063636 0.8252525         1
```