ACHIEVING BALANCE
USING INCIDENCE MATRICES
IN THE DESIGN OF EXPERIMENTS

Acknowledgements: This work was influenced largely by Kansas State Professors Dallas Johnson and Tom Loughin.

Author contact: Mary A. Marion, 934 Marsac Street, Charlottesville, Virginia 22901. E-mail: marymarion@protonmail.com. Voice: 703-851-0446.

Abstract: This paper discusses balance in the construction of good designs. A quantitative measure of the balance is computed using an incidence matrix. Two SAS macros are included.

KEYWORDS: DOE, balance, design matrix, incidence matrix

1.0 Introduction.

An Essential component to the design of experiments is balance of the design. This is to ensure equal representation of treatment combinations in the conduct of the analysis.

2.0 Theory.

Good designs achieve balance in their construction. A quantitative measure of the balance is computed using an incidence matrix. An incidence matrix records whether or not a particular plot or treatment appears in the experiment and if so how many times the treatment appears in combination with each other treatment over all blocks. It is routinely only filled out as an upper triangular matrix.

By considering the balance in the incidence matrix (the number of pairings of treatments in the b blocks of k units) we can prepare better designs. A balanced design maximizes the sum of the elements in the upper triangular portion of the incidence matrix. This is what is usually reported.

3.0 Application.

Balance in a block design occurs when 1) each treatment appears equally often (equal replication) and 2) each pair of treatments appear together in a block equally often. Mathematically, balance is expressed as a positive integer computed as follows:

Let

- b = number of blocks
- k = number of units per block
- t = number of treatments

Then $r = bk/t$ and

λ = # of pairing opportunities / # of treatments to pair with
$= r(k-1) / (t-1)$

The formula for λ does not work with multiple pairs/block.

4.0 Examples

Example 1. Experimental plan for an unbalanced design comparing six treatments in six blocks of four units.

The data is laid out in a table of 4 units per block.

		Block					
		1	2	3	4	5	6
unit	1	C	B	E	A	C	B
	2	E	E	A	C	B	D
	3	F	C	D	D	A	A
	4	D	F	C	F	D	F

$r = bk/t = (6)(4)/6 = 4$
$\lambda = r(k-1)/(t-1) = 4(5-1)/(6-1) = 20/5 = 4$

The corresponding incidence matrix for this data showing the pairing of the treatments is given below.

Incidence Matrix

	B	C	D	E	F
A	2	3	4	1	2
B		2	2	1	2
C			4	3	3
D				2	3
E					2

Example 2. Experimental plan for comparing seven treatments in seven blocks of six units (BIB).

The data is laid out in a table of 6 units per block.

		Blocks						
		1	2	3	4	5	6	7
Unit	1	G	A	B	C	D	E	F
	2	F	G	A	B	C	D	E
	3	A	B	C	D	E	F	G
	4	E	F	G	A	B	C	D
	5	D	E	F	G	A	B	C
	6	C	D	E	F	G	A	B

r = bk/t = (7)(6)/7 = 6

λ = r(k-1)/(t-1)= 6(6-1)/(7-1)=30/6 = 5

The corresponding incidence matrix for this data showing the pairing of the seven treatments within a block is given below.

Incidence Matrix

	B	C	D	E	F	G
A	5	5	5	5	5	5
B		5	5	5	5	5
C			5	5	5	5
D				5	5	5
E					5	5
F						5

Example 3. Experimental plan for comparing seven treatments in seven blocks of five units (PIBD).

The data is laid out in a table of 5 units per block.

		Block						
		1	2	3	4	5	6	7
unit	1	A	B	C	D	E	F	G
	2	G	A	B	C	D	E	F
	3	F	G	A	B	C	D	E
	4	E	F	G	A	B	C	D
	5	D	E	F	G	A	B	C

r = bk/t = (7)(5)/7 = 5

λ = r(k-1)/(t-1)= 5(5-1)/(6-1)=20/5 = 4

The corresponding incidence matrix for this data showing the pairing of the seven treatments is

Incidence Matrix

	B	C	D	E	F	G
A	4	3	3	3	3	4
B		4	3	3	3	3
C			4	3	3	3
D				4	3	3
E					4	3
F						4

Example 4. Experimental plan for comparing six treatments in six blocks of five units (Mead, page 135).

The data is laid out in a table of 5 units per block.

		Block					
		1	2	3	4	5	6
unit	1	A	A	A	A	A	B
	2	B	B	B	B	C	C
	3	C	C	C	D	D	D
	4	D	D	E	E	E	E
	5	E	F	F	F	F	F

$r = bk/t = (6)(5)/6 = 5$
$\lambda = r(k-1)/(t-1) = 5(5-1)/(6-1) = 20/5 = 4$

The corresponding incidence matrix for this data showing the pairing of the six treatments is given below.

Incidence Matrix

	B	C	D	E	F
A	4	4	4	4	4
B		4	4	4	4
C			4	4	4
D				4	4
E					4

5.0 SAS Macro

The SAS macro written to compute the matrix is generic requiring the inputs of

```
r = # of rows in the Design Matrix    = # of units/block
c = # of columns in the Design Matrix = # of blocks
k = # of letters (treatments)         = # of treatments
```

and the design matrix of treatments is in alphanumeric format. The inputs and their outputs to produce these examples are given in the appendix.

The program can easily be altered to input a numerical design matrix by changing from text to numeric input. A second macro is included for such use.

6.0 Acknowledgements.

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. The programs run under SAS 9.3, Enterprise Guide 4.3 and Windows 8.1.

7.0 Textbooks and Individual References

Johnson, Dallas, Kansas State University, Department of Statistics, Dickens Hall, Manhattan, Kansas 66506-0802.

Loughin, Thomas, Kansas State University, Department of Statistics, Dickens Hall, Manhattan, Kansas 66506-0802.

Mead, Roger (1994), The Design of Experiments, Statistical Principles for Practical Application, Cambridge University Press, 40 West 20th Street, New York, New York 10011-4211.

APPENDIX

SAS INPUTS / OUTPUTS

```
options nonumber;


*Example 1;
%let c=6;
%macro DesignMatrix;
%do i=1 %to &c;
   col&i $ %end;
%mend DesignMatrix;

data DataFile1;
input %DesignMatrix;
cards;
C B E A C B
E E A C B D
F C D D A A
D F C F D F
;
%IncidenceMatrixA(DataFile1,4,6,6);
```

```
                    Design_Matrix

   C         B         E         A         C         B
   E         E         A         C         B         D
   F         C         D         D         A         A
   D         F         C         F         D         F


                 Design_Matrix_Numeric

   3          2          5          1          3          2
   5          5          1          3          2          4
   6          3          4          4          1          1
   4          6          3          6          4          6


                    Incidence_Matrix

                  0   2   3   4   1   2
                  0   0   2   2   1   2
                  0   0   0   4   3   3
                  0   0   0   0   2   3
                  0   0   0   0   0   2
                  0   0   0   0   0   0

                   Sum_Elements

                             36
```

```
*Example 2;
%let c=7;
%macro DesignMatrix;
%do i=1 %to &c;
   col&i $ %end;
%mend DesignMatrix;

data DataFile2;
input %DesignMatrix;
cards;
G A B C D E F
F G A B C D E
A B C D E F G
E F G A B C D
D E F G A B C
C D E F G A B
;
%IncidenceMatrixA(DataFile2,6,7,7);
```

```
                          DESIGN_MATRIX

       G        A        B        C        D        E        F
       F        G        A        B        C        D        E
       A        B        C        D        E        F        G
       E        F        G        A        B        C        D
       D        E        F        G        A        B        C
       C        D        E        F        G        A        B


                        DESIGN_MATRIX_NUMERIC

       7         1         2         3         4         5         6
       6         7         1         2         3         4         5
       1         2         3         4         5         6         7
       5         6         7         1         2         3         4
       4         5         6         7         1         2         3
       3         4         5         6         7         1         2


                          INCIDENCE_MATRIX

       0         5         5         5         5         5         5
       0         0         5         5         5         5         5
       0         0         0         5         5         5         5
       0         0         0         0         5         5         5
       0         0         0         0         0         5         5
       0         0         0         0         0         0         5
       0         0         0         0         0         0         0


                            SUM_ELEMENTS

                                 105
```

```
*Example 3;
%let c=7;
%macro DesignMatrix;
%do i=1 %to &c;
   col&i $ %end;
%mend DesignMatrix;

data DataFile3; input %DesignMatrix;
cards;
A B C D E F G
G A B C D E F
F G A B C D E
E F G A B C D
D E F G A B C
;
%IncidenceMatrixA(DataFile3,5,7,7);
```

```
                              DESIGN_MATRIX

       A         B         C         D         E         F         G
       G         A         B         C         D         E         F
       F         G         A         B         C         D         E
       E         F         G         A         B         C         D
       D         E         F         G         A         B         C


                          DESIGN_MATRIX_NUMERIC

       1          2          3          4          5          6          7
       7          1          2          3          4          5          6
       6          7          1          2          3          4          5
       5          6          7          1          2          3          4
       4          5          6          7          1          2          3


                             INCIDENCE_MATRIX

       0          4          3          3          3          3          4
       0          0          4          3          3          3          3
       0          0          0          4          3          3          3
       0          0          0          0          4          3          3
       0          0          0          0          0          4          3
       0          0          0          0          0          0          4
       0          0          0          0          0          0          0


                              SUM_ELEMENTS

                                        70
```

```
*Example 4;

%let c=6;
%macro DesignMatrix;
%do i=1 %to &c;
   col&i $ %end;
%mend DesignMatrix;

data DataFile4; input %DesignMatrix;
cards;
A A A A A B
B B B B C C
C C C D D D
D D E E E E
E F F F F F
;
%IncidenceMatrixA(DataFile4,5,6,6);
```

```
              DESIGN_MATRIX

    A    A    A    A    A    B
    B    B    B    B    C    C
    C    C    C    D    D    D
    D    D    E    E    E    E
    E    F    F    F    F    F


            DESIGN_MATRIX_NUMERIC

    1    1    1    1    1    2
    2    2    2    2    3    3
    3    3    3    4    4    4
    4    4    5    5    5    5
    5    6    6    6    6    6


              INCIDENCE_MATRIX

    0    4    4    4    4    4
    0    0    4    4    4    4
    0    0    0    4    4    4
    0    0    0    0    4    4
    0    0    0    0    0    4
    0    0    0    0    0    0

              SUM_ELEMENTS

                   60
```

```
*Example, Mead (page 160);

%start(EMF,L,n,Mead160);

%let c=7;
%macro DesignMatrix;
%do i=1 %to &c;
   col&i %end;
%mend DesignMatrix;

data DataFile5;
input %DesignMatrix;
cards;
1   19  37   7  13   4  10
2   20  38   8  14   5  11
3   21  39   9  15   6  12
4   22  40  10  16  28  19
5   23  41  11  17  29  20
6   24  42  12  18  30  21
7   25   1  25  31   7  16
8   26   2  26  32   8  17
9   27   3  27  33   9  18
10  28   4  28  34  22  40
11  29   5  29  35  23  41
12  30   6  30  36  24  42
13  31  19  37   1  13  31
14  32  20  38   2  14  32
15  33  21  39   3  15  33
16  34  22  40  25  37  34
17  35  23  41  26  38  35
18  36  24  42  27  39  36
;

options ls=256;
%IncidenceMatrixN(DataFile5,18,7,42);
```

```
Today's date is 06/24/15 and the time is 22:17

Input File:  Mead160.sas
Log File:    Mead160.log
Output File: Mead160.lst
Author:      Mary A. Marion

                    Design_Matrix_Numeric
        1        19        37         7        13         4        10
        2        20        38         8        14         5        11
        3        21        39         9        15         6        12
        4        22        40        10        16        28        19
        5        23        41        11        17        29        20
        6        24        42        12        18        30        21
        7        25         1        25        31         7        16
        8        26         2        26        32         8        17
        9        27         3        27        33         9        18
       10        28         4        28        34        22        40
       11        29         5        29        35        23        41
       12        30         6        30        36        24        42
       13        31        19        37         1        13        31
       14        32        20        38         2        14        32
       15        33        21        39         3        15        33
       16        34        22        40        25        37        34
       17        35        23        41        26        38        35
       18        36        24        42        27        39        36
```

```
                    Incidence_Matrix
0 3 3 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 3 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 3 3 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 0 0 0 1 1 1 0 0 0 0 0 0 2 2 2 1 1 1
0 0 0 0 0 3 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 0 0 0 1 1 1 0 0 0 0 0 0 2 2 2 1 1 1
0 0 0 0 0 0 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 0 0 0 1 1 1 0 0 0 0 0 0 2 2 2 1 1 1
0 0 0 0 0 0 0 3 3 2 2 2 2 2 2 1 1 1 0 0 0 1 1 1 1 1 1 2 2 2 0 0 0 0 0 0 2 2 2 1 1 1
0 0 0 0 0 0 0 0 3 2 2 2 2 2 2 1 1 1 0 0 0 1 1 1 1 1 1 2 2 2 0 0 0 0 0 0 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 1 1 1 0 0 0 1 1 1 1 1 1 2 2 2 0 0 0 0 0 0 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 3 3 1 1 1 2 2 2 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 0 0 3 1 1 1 2 2 2 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 2 2 2 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 2 2 2 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 2 2 2 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 1 1 1 0 0 0 1 1 1 0 0 0 2 2 2 2 2 2 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 1 1 1 0 0 0 1 1 1 0 0 0 2 2 2 2 2 2 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 2 2 2 2 2 2 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 1 1 1 1 1 1 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 1 1 1 1 1 1 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 3 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 3 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 3 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 2 2 2
```

```
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 2 2 2
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Sum_Elements
        1071
```

SAS Source Code -Alphanumeric Input

```
%macro IncidenceMatrixA(DataFile,r,c,k);
/* © COPYRIGHT 2001 BY Mary A. Marion */
proc iml;

/* PARAMETER INITIALIZATION
     r= # of rows in the Design Matrix
     c= # of columns in the Design Matrix
     k= # of letters ( treatments )

    INPUT:       Alphanumeric Design Matrix
    LIMITATIONS: 20 Treatments  */

   use &DataFile;
   read all var _char_ into Design_Matrix; close &DataFile;
   print Design_Matrix;

/* Change letters to numbers and put into iml matrix Design_Matrix_Numeric */
   Design_Matrix_Numeric=j(nrow(Design_Matrix),ncol(Design_Matrix),.);
   do i=1 to &r;
      do j=1 to &c;
         if Design_Matrix[i,j]='A'      then Design_Matrix_Numeric[i,j]=1;
         else if Design_Matrix[i,j]='B' then Design_Matrix_Numeric[i,j]=2;
         else if Design_Matrix[i,j]='C' then Design_Matrix_Numeric[i,j]=3;
         else if Design_Matrix[i,j]='D' then Design_Matrix_Numeric[i,j]=4;
         else if Design_Matrix[i,j]='E' then Design_Matrix_Numeric[i,j]=5;
         else if Design_Matrix[i,j]='F' then Design_Matrix_Numeric[i,j]=6;
         else if Design_Matrix[i,j]='G' then Design_Matrix_Numeric[i,j]=7;
         else if Design_Matrix[i,j]='H' then Design_Matrix_Numeric[i,j]=8;
         else if Design_Matrix[i,j]='I' then Design_Matrix_Numeric[i,j]=9;
         else if Design_Matrix[i,j]='J' then Design_Matrix_Numeric[i,j]=10;
         else if Design_Matrix[i,j]='K' then Design_Matrix_Numeric[i,j]=11;
         else if Design_Matrix[i,j]='L' then Design_Matrix_Numeric[i,j]=12;
         else if Design_Matrix[i,j]='M' then Design_Matrix_Numeric[i,j]=13;
         else if Design_Matrix[i,j]='N' then Design_Matrix_Numeric[i,j]=14;
         else if Design_Matrix[i,j]='O' then Design_Matrix_Numeric[i,j]=15;
         else if Design_Matrix[i,j]='P' then Design_Matrix_Numeric[i,j]=16;
         else if Design_Matrix[i,j]='Q' then Design_Matrix_Numeric[i,j]=17;
         else if Design_Matrix[i,j]='R' then Design_Matrix_Numeric[i,j]=18;
         else if Design_Matrix[i,j]='S' then Design_Matrix_Numeric[i,j]=19;
         else if Design_Matrix[i,j]='T' then Design_Matrix_Numeric[i,j]=20;
     end;
   end;
   print Design_Matrix_Numeric;

 /* Compute the Incidence Matrix */
   k=&k;
   Incidence_Matrix=j(k,k,0);
   count1=j(k,k,0); count2=j(k,k,0);

   do   j=1 to &c by 1;
      do  jj=1 to k-1 by 1;
```

```
          do jjj=2 to k by 1;
             if jj<jjj then do;
                do   i=1 to &r by 1;
                if Design_Matrix_Numeric[i,j]=jj  then
                      count1[jj,jjj]=count1[jj,jjj]+1;
                if Design_Matrix_Numeric[i,j]=jjj then
                      count2[jj,jjj]=count2[jj,jjj]+1;
                end;
          end;

      do cjj=1 to k-1 by 1;
         do cjjj=2 to k by 1;
            if count2[cjj,cjjj] < count1[cjj,cjjj]  then do;
            count1[cjj,cjjj]=count2[cjj,cjjj]; end;
          end;
      end;

      Incidence_Matrix[jj,jjj] = Incidence_Matrix[jj,jjj] + count1[jj,jjj];
      count1[jj,jjj]=0; count2[jj,jjj]=0;
            end;
         end;
      end;

      Sum_Elements=sum(Incidence_Matrix);
      print Incidence_Matrix[format=3.0], Sum_Elements;

   quit;
   %mend;
```

SAS Source Code -Numeric Input

```
%macro IncidenceMatrixN(DataFile,r,c,k);
/* © COPYRIGHT 2001 BY Mary A. Marion */
proc iml;

/* PARAMETER INITIALIZATION
     r= # of rows in the Design Matrix
     c= # of columns in the Design Matrix
     k= # of treatments
   INPUT: Numeric Design Matrix */

   use &DataFile;
   Design_Matrix_Numeric=j(&r,&c);
   read  all into Design_Matrix_Numeric; close &DataFile;
   print Design_Matrix_Numeric;

 /* Compute the Incidence Matrix */
   k=&k;
   Incidence_Matrix=j(k,k,0);
   count1=j(k,k,0); count2=j(k,k,0);

   do   j=1 to &c by 1;
      do  jj=1 to k-1 by 1;
         do jjj=2 to k by 1;
            if jj<jjj then do;
               do   i=1 to &r by 1;
               if Design_Matrix_Numeric[i,j]=jj  then
count1[jj,jjj]=count1[jj,jjj]+1;
               if Design_Matrix_Numeric[i,j]=jjj then
count2[jj,jjj]=count2[jj,jjj]+1;
               end;
         end;

   do cjj=1 to k-1 by 1;
      do cjjj=2 to k by 1;
         if count2[cjj,cjjj] < count1[cjj,cjjj]  then do;
         count1[cjj,cjjj]=count2[cjj,cjjj]; end;
        end;
   end;

   Incidence_Matrix[jj,jjj] = Incidence_Matrix[jj,jjj] + count1[jj,jjj];
   count1[jj,jjj]=0; count2[jj,jjj]=0;

         end;
      end;
   end;

   Sum_Elements=sum(Incidence_Matrix);
   print Incidence_Matrix[format=3.0], Sum_Elements;

quit;
%mend;
```

PROGRAM DOCUMENTATION

Given: generic incidence matrix allowing for multiple repetitions of treatments

Example: Two (1,2) pairs

Units/Block

1				
1				
1				
2				
3				
1				
2				

Subscript Methodology --- > Counting Strategy

	Treatment
JJ	1
JJJ	2
	3
	…
	k

Count1	Count2
1	2
1	3
2	3

1   2

Count Matrix 1 keeps track of count number of 1st values in treatment pair.

	1	2	3
1	0	4	4
2	0	0	2
3	0	0	0

Count matrix 2 keeps track of count number of 2nd values in treatment pair.

	1	2	3
1	0	2	1
2	0	0	1
3	0	0	0

Procedure implemented by the program.

- Select a treatment pair
- Count number of values in incidence matrix that match 1st value in treatment pair in count 1 matrix.
  Count number of values in incidence matrix that match 2nd value of pair
- Get treatment pair count from smallest value from pair in count1 and count2 matrix
- Move pair count to incidence matrix.