Quick Guide of SAS/IML Commands

```
Define a Matrix  M={1 2 3,   Element Identification  a =M[2,3]  a=6
                    4 5 6,   Row Identification      ar=M[2,]   ar = 4 5 6
                    7 8 9};  Column Identification   ac=M[,1]   ac= 1
                                                                    4
                                                                    7
```

Operators

```
  Addition               +   Matrices must conform  Can work with a scalar
  Subtraction            -   Matrices must conform  Can work with a scalar
  Matrix Multiplication  *   Matrices must conform  Can work with a scalar
  Element Multiplication #   Matrices must conform  Can work with a scalar
  Element Division       /   Matrices must conform  Can work with a scalar
  Matrix Power           **
  Element Power          ##

  Less than               <
  Greater than            >
  Equal to                =
  Less than or equal to   <=
  Greater than or equal   >=
  Not equal to            ^=

Reduction Operators
  Addition                +
  Subtraction             -
  Maximum                 <>
  Minimum                 ><
  Index of Maximum        <:>
  Index of Maximum        >:<
  Mean                     :
  Sum of Squares          ##

  Transpose                 `
  Horizontal Concatenation  ||
  Vertical Concatenation    //

Row Summation   ars=M[,+];  ars =  6     Row Means  arm=M[,:]; arm = 2
                                  15                                 5
                                  24                                 8

Column Summation  acs=M[+,];  acs =  12  15  18
Column Means      acm=M[:,];  acm =   4   5   6
Column SS         acq=M[##,]; acq=   66  93 126

Row SS            arq=M[,##]; arq=   14  Total SS   mss=ssq(M)= 285
                                     77
                                    194

Number of rows nr=nrow(M); nr = 3; Number of columns nc=ncol(M); nc = 3;
Absolute Value G={-1 4 –5};    A=abs(G)={1 4 5};
```

Pvalue for F pf=1-probf(F,dfn,dfd);Palue for χ^2 = pchi=1-probchi(X,df);

```
Create an N x P matrix of Gaussian deviates E=rannor(j(N,P,0));
                          Uniform deviates  E=ranuni(j(N,P,0));
```

```
Define a Matrix  M={1 2 3,   N={8 2 3,   Create an R x C matrix of
                    4 5 6,      4 9 6,   zeros  ZED=j(R,C,0);
                    7 8 9};     7 1 5};  ones   ONE=j(R,C,1);
                 B={0 1,                 Create an N-dimensional
                    2 3};                 Identity Matrix IN = I(N);

Matrix                                 Scalar
Addition S=M+N;  S={ 9  4  6,          Addition   S=N+3; S={ 11  3  4,
                     8 14 12,                                 5 12  9,
                    14  9 14}                                10  4  8}

Matrix                                 Scalar
Subtraction S=M-N;  S={-7  0  0,       Subtraction   S=N-2; S={  6  0  1,
                        0 -4  0,                                 2  7  4,
                        0  7  4}                                 5 -1  3}

Matrix                                    Element
Multiplication S=M*N; N={ 37  23  30,     Multiplication  S=M#N; S={  8  4  9,
                          94  59  72,                or S=N#M;       16 45 36,
                         151  95 114}                                49  8 45}

Matrix                                     Scalar
Division S=M/N;  S={0.125  1.000  1.000,   Division S=N/4; S={2.00  0.50  0.75,
                    1.000  0.556  1.000,                      1.00  2.25  1.50,
                    1.000  8.000  1.800}                      1.75  0.25  1.25}

Matrix                                    Element
Power  S=M**2; S={ 30  36  42,            Power  S=M##2;  S={ 1  4  9,
   or S=M*M;       66  81  96,                or S=M#M;      16  25 36,
                  102 126 150}                               49  64 81}

Matrix                                         Element Square Root
Inversion S=inv(N);  S={ .218  -.039  -.084,     S=N##.5; S={2.83  1.41  1.73,
                         .123   .106  -.201, or S=sqrt(N);  2.00  3.00  2.45,
                        -.330   .034   .358}                2.65  1.00  2.24}

Concatenation                      Vertical  S=M//N;  S={1 2 3,
Horizontal    S=M||N;                                    4 5 6,
          S = {1 2 3 8 2 3,                              7 8 9,
               4 5 6 4 9 6,                              8 2 3,
               7 8 9 7 1 5}                              4 9 6,
                                                         7 1 5}

Kronecker Product S=B@M;            S= I(2)@N;
S=    0    0    0     1    2    3   S= 8   2   3     0   0   0
      0    0    0     4    5    6      4   9   6     0   0   0
      0    0    0     7    8    9      7   1   5     0   0   0
      2    4    6     3    6    9      0   0   0     8   2   3
      8   10   12    12   15   18      0   0   0     4   9   6
     14   16   18    21   24   27      0   0   0     7   1   5
```

Using SAS Data sets in IML

```
data jj;
input ID Y X1 X2 X3;
cards;
1 3 4 5 1
2 8 5 4 1
3 9 2 1 1
4 7 6 4 0
5 5 3 6 0
6 4 7 3 0
;proc reg data=jj;
model y = x1 x2 x3;run;
```

The REG Procedure
Model: MODEL1
Dependent Variable: Y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	11.63796	3.87932	0.47	0.7320
Error	2	16.36204	8.18102		
Corrected Total	5	28.00000			

Root MSE	2.86025	R-Square	0.4156
Dependent Mean	6.00000	Adj R-Sq	-0.4609
Coeff Var	47.67080		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	10.46826	5.83133	1.80	0.2145
X1	1	-0.33312	0.78552	-0.42	0.7128
X2	1	-0.77498	0.78552	-0.99	0.4278
X3	1	0.00314	2.81768	0.00	0.9992

```
proc iml;
use jj;
read all var{y} into Y;
read all var{X1 X2 X3} into XM;
read point 2 var{Y} into Y8;
read point 6 var{ID} into N6;
N=nrow(Y);
X=(j(N,1,1))||XM;** Create a vector with N=6 rows, 1 column, of ones **;
B=(inv(X`*X))*(X`*Y);
print Y X Y8 N6;
print B;
```

Y	X				Y8	N6
3	1	4	5	1	8	6
8	1	5	4	1		
9	1	2	1	1		
7	1	6	4	0		
5	1	3	6	0		
4	1	7	3	0		

B
10.468259
-0.333124
-0.774984
0.003143