Introduction to SAS IML: Interactive Matrix Language

Bios 524: Biostatistical Computing
Getting Help!

- Use the SAS OnLine Documentation for help on SAS IML
  (click the icons – they are links to VCU’s copy of SAS OnlineDoc).

- Language Reference
What is SAS IML?

- From the OnLine Doc: SAS/IML software
  - is a programming language.
  - operates on matrices.
  - possesses a powerful vocabulary of operators.
  - uses operators that apply to entire matrices.
  - is interactive.
  - is dynamic.
  - processes data.
  - produces graphics.
A Simple Example

Solve this system of linear equations:

\[ \begin{align*}
3x + 2y - 4z &= 11 \\
5x - 4y &= 9 \\
3y + 10z &= 42
\end{align*} \]

In matrix terms:

\[ MA = B \]

Solve for \( A \).
A Simple Example: IML Solution

```
proc iml;
M={3 2 -4, 5 -4 0, 0 3 10};
B={11, 9, 42};
A=solve(M,B);
```

Result:

```
A=
 5
 4
 3
```
Creating Matrices

- Assigning vectors and matrices
- Creating matrices with functions and operators
- Special Matrices
  - Identity
  - Constant
  - Diagonal
Matrix Operators

- Addition, subtraction, negation (+, –)
- Multiplication
  - (*)
    - Matrix multiplication
    - Scalar multiplication
  - Element-wise multiplication (#)
- Division (/)
  - Matrix/Matrix
  - Matrix/scalar or scalar/Matrix
  - Scalar/scalar
More Matrix Operators

- **Power**
  - Based on matrix multiplication (**\**)
    - Example: \( M \ast M \ast M \) same as \( M \ast \ast 3 \).
  - Based on element-wise multiplication (##)
    - Example: \( M \# M \# M \) same as \( M \# \# \# 3 \).

- **Concatenation**
  - Horizontal (||)
  - Vertical (//)
Comparison Operators

- **Comparisons** (<, ≤, >, ≥, ^=)
  - Matrices are compared element by element. The result is a matrix of 0’s and 1’s, with a 1 indicating that the corresponding element comparison is true.
  - If the comparison is used in a conditional statement, then all element comparisons must be true (=1) for the conditional statement to execute.

- **Logical Operators** (&, |, ^)
  - Element-wise comparisons
    - (&) Element is 1 if both corresponding elements are nonzero.
    - (|) Element is 1 if at least one corresponding element is nonzero.
  - Not prefix (^): Converts zeros to 1 and nonzeros to 0.
Matrix Subscripts

- Select a single element: $X[2,3]$
- Select a row: $X[2,]$
- Select a column: $X[,3]$
- Select a submatrix
  - $X[\{1\ 2\ 3\},\{2\ 3\}]$ or $X[1:3,2:3]$
- Assign values to submatrix
  - $X[2,3]=0$
- Subscripts may also be functions or expressions: $M[(M<0)]=0$
Subscript Reduction Operators

- Operators may be used in place of subscripts that reduce the matrix by operating on certain elements.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>addition</td>
</tr>
<tr>
<td>#</td>
<td>multiplication</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>maximum</td>
</tr>
<tr>
<td>&gt;:&lt;</td>
<td>minimum</td>
</tr>
<tr>
<td>&lt;::&gt;</td>
<td>index of maximum</td>
</tr>
<tr>
<td>&gt;::&lt;</td>
<td>index of minimum</td>
</tr>
<tr>
<td>:</td>
<td>mean (different from the MATRIX procedure)</td>
</tr>
<tr>
<td>##</td>
<td>sum of squares</td>
</tr>
</tbody>
</table>
Exercise 1

- Assign these two matrices to $A$ and $B$, respectively:

$A = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 7 \\ 2 & 3 & 3 \end{pmatrix}$

$B = \begin{pmatrix} 5 & 6 \\ 3 & 4 \\ 4 & 6 \end{pmatrix}$

- Find the sum of all values in $A$.
- Find the sum of each column in $B$.
- Find the matrix product, $AB$.
- Find the value of $C = 6(A_1 A_2) - 5B$.
  - Find the sum of the positive elements of $C$.
  - Set the negative values of $C$ to zero.
- Solve $AU = B$ for $U$ (a $3 \times 2$ matrix).
Using IML with SAS Datasets

- Create matrices from the variables and observations of a SAS data set in several ways.
  - Create a column vector for each data set variable.
  - Create a matrix where columns correspond to data set variables.
  - You can use all the observations in a data set or use a subset of them.

- Create a SAS data set from a matrix.
  - The columns correspond to data set variables and the rows correspond to observations.
  - When reading a SAS data set, you can read any number of observations into a matrix either sequentially, directly by record number, or conditionally.
Open a SAS Data Set for Reading into a Matrix

- **Read only:**
  
  ```
  USE SAS-data-set < VAR operand > < WHERE(expression) > < NOBS name > ;
  ```

  - **SAS-data-set**
    
    can be specified with a one-word name (for example, A) or a two-word name (for example, SASUSER.A).  
    
    *operand*
    
    selects a set of variables.

  - **expression**
    
    is evaluated for being true or false.  
    
    Selects a set of observations.

  - **name**
    
    is the name of a variable to contain the number of observations.
Read Data from a
SAS Data Set into a Matrix

Read from the opened SAS data set:

READ < range > < VAR operand > < WHERE(expression) > < INTO name > ;

- range
  specifies a range of observations.

- operand
  selects a set of variables.

- expression
  is evaluated for being true or false. Selects a set of observations.

- name
  names a target matrix for the data.
Create a SAS Data Set
from a Vectors

Create a new SAS data set:
CREATE SAS-data-set <VAR operand>;

operand
selects a set of variables.

- A literal containing variable names
- the name of a matrix containing variable names
- an expression in parentheses yielding variable names
- one of the keywords described below:
  - _ALL_
  - for all variables
  - _CHAR_ for all character variables
  - _NUM_ for all numeric variables
Create a SAS Data Set from a Matrix

Create a new SAS data set:

```sas
CREATE SAS-data-set <From matrix-name>
  <[COLNAME=column-name ROWNAME=row-name]>;
```

- **column-name**
  - is a character matrix or quoted literal containing descriptive names to associate with data set variables.

- **row-name**
  - is a character matrix or quoted literal containing descriptive names to associate with observations on the data set.
Data Management Commands

Look these up, for example:

- **Append**
  adds observations to the end of a SAS data set

- **Show Contents**
  shows contents of the current input SAS data set

- **Show Datasets**
  shows SAS data sets currently open

- **Close**
  closes a SAS data set

- **Setin**
  selects an open SAS data set for input
Exercise 2

Use IML to place the number of FTE physicians in the clinics into a vector.

- Find the average FTE.
- Find the maximum FTE.
- Find the smallest FTE that is greater than the mean.
  - Challenge: Find all clinics corresponding to this value.
  - Challenge: Print the clinic names using $CLINID format.

Note: Use the libref LIBRARY, since CLINICS uses custom formats.
I ML Programming

- Structured programming
- Modules
- Functions and call routines
- IMLMLLIB: Module Library
Regression Example

- Model

\[ y = \alpha + \beta_1 X + \beta_2 X^2 + \varepsilon \]

- X: 1, 2, 3, 4, 5
- Y: 1, 5, 9, 23, 36
- Regress Y on X
  - Find estimates of \( \alpha, \beta_1, \beta_2 \)
  - Find predicted values of Y.
Exercise 3

Convert this table of means to a matrix:

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>13</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>10</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Find values of $\mu$, $\alpha_i$, $\beta_j$, $\gamma_{ij}$ where

$$E(Y_{ij}) = \mu + \alpha_i + \beta_j + \gamma_{ij}$$