

Chapter 1

Introduction to SAS/IML Software

Chapter Contents

OVERVIEW	7
SAS/IML SOFTWARE: AN OVERVIEW	7
AN INTRODUCTORY INTERACTIVE SESSION	9
PROC IML STATEMENT	9

Chapter 1

Introduction to SAS/IML Software

Overview

SAS/IML software gives you access to a powerful and flexible programming language (Interactive **M**atrix **L**anguage) in a dynamic, interactive environment. The fundamental object of the language is a data matrix. You can use SAS/IML software interactively (at the statement level) to see results immediately, or you can store statements in a module and execute them later. The programming is dynamic because necessary activities such as memory allocation and dimensioning of matrices are performed automatically.

SAS/IML software is powerful. You can access built-in operators and call routines to perform complex tasks such as matrix inversion or eigenvector generation. You can define your own functions and subroutines using SAS/IML modules. You can operate on a single value or take advantage of matrix operators to perform operations on an entire data matrix. For example, the statement

```
x=x+1;
```

can be used to add 1 to a single value **X**, or to add 1 to all elements of a matrix **X**.

You have access to a wide choice of data management commands. You can read, create, and update SAS data sets from inside SAS/IML software without ever using the DATA step. For example, reading a SAS data set to get phone numbers for all individuals whose last name begins with “Smith” is easy:

```
read all var{phone} where(lastname="Smith");
```

The result is a matrix **PHONE** of phone numbers.

SAS/IML Software: An Overview

SAS/IML software is a programming language.

You can program easily and efficiently with the many features for arithmetic and character expressions in SAS/IML software. You have access to a wide range of built-in subroutines designed to make your programming fast, easy, and efficient. Because SAS/IML software is part of the SAS System, you can access SAS data sets or external files with an extensive set of data processing commands for data input and output, and you can edit existing SAS data sets or create new ones.

SAS/IML software has a complete set of control statements, such as **DO/END**, **START/FINISH**, iterative **DO**, **IF-THEN/ELSE**, **GOTO**, **LINK**, **PAUSE**, and **STOP**, giving you all of the commands necessary for execution control and program modularization.

SAS/IML software operates on matrices.

While most programming languages deal with single data elements, the fundamental data element with SAS/IML software is the matrix, a two-dimensional (row \times column) array of numeric or character values.

SAS/IML software possesses a powerful vocabulary of operators.

You can access built-in matrix operations that require calls to math-library subroutines in other languages. You have access to many operators, functions, and CALL subroutines.

SAS/IML software uses operators that apply to entire matrices.

You can add elements of the matrices **A** and **B** with the expression $\mathbf{A} + \mathbf{B}$. You can perform matrix multiplication with the expression $\mathbf{A} * \mathbf{B}$ and perform elementwise multiplication with the expression $\mathbf{A} \# \mathbf{B}$.

SAS/IML software is interactive.

You can execute a command as soon as you enter it, or you can collect commands in a module to execute later. When you execute a command, you see the results immediately. You can interact with an executing module by programming IML to pause, enabling you to enter additional statements before continuing execution.

SAS/IML software is dynamic.

You do not need to declare, dimension, and allocate storage for a data matrix. SAS/IML software does this automatically. You can change the dimension or type of a matrix at any time. You can open multiple files or access many libraries. You can reset options or replace modules at any time.

SAS/IML software processes data.

You can read all observations or read conditionally selected observations from a SAS data set into a matrix, creating either multiple vectors (one for each variable in the data set) or a matrix that contains a column for each data set variable. You can create a new SAS data set, or you can edit or append observations to an existing SAS data set.

SAS/IML software produces graphics.

You have access to a wide range of graphics commands, enabling you to visually explore relationships in data.

An Introductory Interactive Session

Here is a simple introductory session that uses SAS/IML software to estimate the square root of a number, accurate to three decimal places. In this session, you define a function module named APPROX to perform the calculations and return the approximation. You then call APPROX to estimate the square root of several numbers given in a matrix literal (enclosed in braces), and you print the results.

Throughout the session, the right angle brackets (>) indicate statements that you submit; responses from IML follow.

```

> proc iml;                                /* begin IML session */

      IML Ready

> start approx(x);                          /* begin module */
>   y=1;                                    /* initialize y */
>   do until(w<1e-3);                       /* begin do loop */
>     z=y;                                  /* set z=y */
>     y=.5#(z+x/z);                         /* estimate square root */
>     w=abs(y-z);                          /* compute change in estimate */
>   end;                                    /* end do loop */
>   return(y);                              /* return approximation */
> finish approx;                            /* end module */

      NOTE: Module APPROX defined.

> t=approx({3,5,7,9});                     /* call function APPROX */
> print t;                                  /* print matrix */

                                     T
                                     1.7320508
                                     2.236068
                                     2.6457513
                                     3

> quit;

      Exiting IML

```

PROC IML Statement

PROC IML <SYMSIZE=*n1*> <WORKSIZE=*n2*>;

where *n1* and *n2* are specified in kilobytes.

The SYMSIZE= and WORKSIZE= options in the PROC IML statement give you control over the size of memory allocated to the symbol space and the size of each extent of workspace. If you do not specify these options, PROC IML uses host dependent defaults.

Generally, you do not need to be concerned with the details of memory usage because memory allocation is done automatically. For special situations, however, see [Memory and Workspace](#).