

# A Primer on Using SAS

James L. Schmidhammer  
University of Tennessee

## SAS Programs

- A sequence of statements executed in order.

## SAS Program Statements

- Every SAS statement ends with a semicolon(;).
- Letters may be uppercase or lowercase. SAS statements are case insensitive.
- Statements can span multiple lines. SAS simply searches for the next occurring semicolon(;), and uses this to identify the end of the statement.
- Multiple statements can be on one line.
- Statements may start in any column.

## SAS Data Types

Two kinds of data types:

- Numeric data
  - Variables contain numerical values (numbers):  
`X = 3;`
- Character data
  - Variables contain text:  
`City = 'Knoxville';`

## Missing Data

- Missing numeric data is represented by a single period(.):

```
Y = .;
```

- Missing character data is represented by blanks

```
Middle_Name = ' ';
```

## SAS Variable Names

- 1 – 32 characters [letters, numerals and the underscore(\_)] in length.
- Must begin with a letter or an underscore(\_).

## SAS Comments

Two possible forms:

- Comments start with an asterisk(\*) and end with a semicolon(;).

```
*This is a comment;
```

- An alternative allowable form is:

```
/* This is a comment */
```

## SAS Programs

Two principal components:

- DATA step(s)
- PROC step(s)

## SAS DATA Step

- Begins with a DATA statement
- Reads and modifies data
- Creates a SAS data set

## SAS PROC Steps

### SAS Procedures

- Begin with a PROC statement
- Perform specific analyses or functions
- Produce results to be viewed or printed

## Generic SAS Program: SAS statements and data all in one file

```
OPTIONS options;

TITLE 'title';
DATA datasetname;
  INPUT varname1 varname2 ... ;
  (data transformations, expressions, and functions go here)
  DATALINES;
  data (for first case)
  data (for second case)
  .
  .
  .
  data (for last case)
  ;
RUN;

PROC ... ; RUN;
PROC ... ; RUN; QUIT;
```

## Generic SAS Program: SAS statements stored in a file; data stored in a separate ASCII file named d:\Example.txt

```
OPTIONS options;

TITLE 'title';

DATA datasetname;
  INFILE 'd:\EXAMPLE.TXT';
  INPUT varname1 varname2 ... ;
  (data transformations, expressions, and functions go here)
RUN;

PROC ... ; RUN;

PROC ... ; RUN; QUIT;
```

**Generic SAS Program:**  
SAS statements stored in a file;  
data stored in a separate XPort SAS dataset  
named d:\Example.xpt

```
OPTIONS options;  
TITLE 'title';  
LIBNAME L XPORT 'd:\';  
  
PROC CONTENTS DATA=L.EXAMPLE;  
  
DATA datasetname;  
  SET L.EXAMPLE;  
  (data transformations, expressions, and functions go here)  
RUN;  
  
PROC CONTENTS DATA=datasetname;  
PROC ... ; RUN;  
PROC ... ; RUN; QUIT;
```

**Generic SAS Program:**  
SAS statements stored in a file;  
data stored in a separate SAS dataset named  
d:\Example.sas7bdat

```
OPTIONS options;  
TITLE 'title';  
LIBNAME L 'd:\';  
  
PROC CONTENTS DATA=L.EXAMPLE;  
  
DATA datasetname;  
  SET L.EXAMPLE;  
  (data transformations, expressions, and functions go here)  
RUN;  
  
PROC CONTENTS DATA=datasetname;  
PROC ... ; RUN;  
PROC ... ; RUN; QUIT;
```

## Autoexec.sas file

- In a typical installation, the executable file `sas.exe` resides in the following subdirectory:  
    `c:\Program Files\SAS\SASFoundation\9.2\`
- Create a file named  
    `c:\Program Files\SAS\SASFoundation\9.2\autoexec.sas`
- Any SAS commands contained in this file will be automatically executed every time that SAS is started.

## Example of Autoexec.sas

```
options pageno=1 nodate;
```

```
filename statmacs 'C:\Documents and Settings\JLS\My Documents\My SAS Files\Macros\';  
options maautosource sasautos=(statmacs, sasautos);
```

```
libname library 'C:\Documents and Settings\JLS\My Documents\My SAS Files\Datasets\';  
libname Mult 'C:\Documents and Settings\JLS\My Documents\My SAS Files\Datasets';
```