

SAS/ASSIST Software for Beginners

Sally Muller

Arturo Barrios

University of North Carolina, Chapel Hill, NC

Abstract

Four years ago we began teaching a two-hour introduction to SAS/ASSIST. Most of the hundreds of students have been SAS novices with limited computing knowledge of any sort. SAS/ASSIST has always provided a simple and non-threatening, yet very capable, tool for using SAS and learning the language.

Introduction

When facing a task as daunting as using the SAS System for the first time, you want to first consider your options. One of these options is the SAS/ASSIST product -- a "point and click" interface that offers immediate access to the SAS System. It has two distinct advantages over conventional methods of using the SAS System.

First, although ASSIST does not require that you write a single line of SAS code, it's a great way of learning how to write SAS code. This is because for each task that you accomplish in ASSIST, ASSIST generates a fully documented SAS program. By saving this code, a very straightforward process, you can study it later and considerably leverage your learning of the SAS programming language.

Second, the up-front time cost of learning to use the SAS System with ASSIST is significantly shorter than learning to use the SAS System with conventional methods. When you use ASSIST you go straight to the work at hand and get results immediately.

This tutorial covers the role of your data in the SAS System and how to translate raw data to a SAS file. We will also show you other common tasks that can be accomplished with ASSIST. We will particularly emphasize how to save the SAS code of every task you perform,

as a means of helping you learn to write SAS programs.

Your Data - The Center of the SAS Universe

Flat Files

Frequently your data is initially in a "flat file". This file may also be referred to as a "raw," "external," or a "text" file. Flat files can be ASCII files, EBCDIC files, and other kind of files. Their one distinguishing characteristic is that they are not SAS files.

[In this tutorial we will refer to these files as "flat" files since that is how SAS/ASSIST refers to them.]

SAS Data Sets

Getting your data into a form that the SAS System can process can be one of the most challenging tasks you face. If you convert your data to a SAS data set, however, you may run all of the utilities (called "PROCedures") of the SAS System against that SAS data set, including all of the statistical procedures.

One of the easiest ways to think of a SAS data set is as a spreadsheet. Like a spreadsheet a SAS data set has a rectangular form. Data values in a row are called "observations", and those in a column are called "variables". A variable may take on either numeric or character value.

SAS Libraries

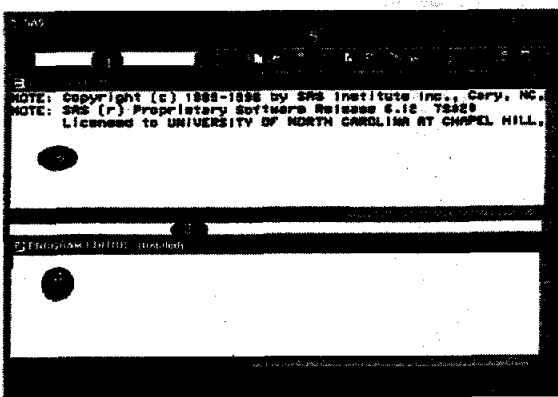
A legacy from its early mainframe days, the SAS Library is a difficult concept to grasp. The important thing to remember is that *every SAS data set is a member of a SAS Library*. And that every SAS library contains one or more SAS files. On a particular operating system the SAS library may correspond to a subdirectory (DOS, Windows, OS/2, VMS, and UNIX) or a file (MVS, CMS.)

Exercise 1 – Starting SAS/ASSIST

SAS/ASSIST does not work in batch mode. Therefore to invoke ASSIST you must be in the SAS Display Manager System (DMS). The Display Manager System is the preferred interactive SAS interface. How you invoke DMS depends on your release of the SAS System, your operating system, and other factors. Usually typing SAS at a command prompt or clicking on a SAS icon in your desktop will invoke the DMS.

[In this tutorial we will be using the SAS System under the Microsoft Windows environment, so our exercises may look slightly different from your own.]

Regardless of how you invoke DMS, when your session begins it will look something like this:



Typically four windows will be open:

1. Command window --- A small window for entering SAS commands.
2. Log --- From this window you can monitor the progress of executing jobs. Errors and warnings messages are reported here. (When you first start a session the window reports current information about the version of SAS that you are running and the TS number, which is the "maintenance" release number, etc.)
3. Output --- The Output window allows you to view the results of a SAS procedure. Although the Log and Program Editor windows mostly cover this window, it enlarges when you have output to see.

4. Program Editor --- Is a window running the SAS editor, which is a simple text editor for creating text files. You may use this editor for writing your SAS programs or you may retrieve programs created in another plain text editor into this window.

Besides the windows you have these important tools for using DMS and ASSIST:

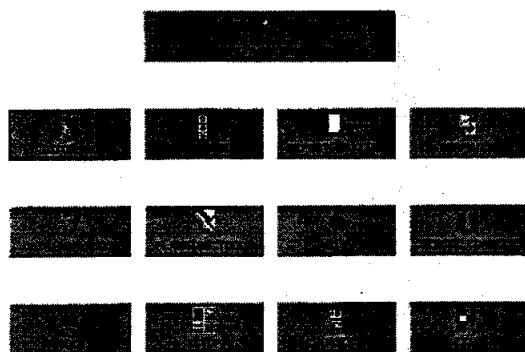
5. Pull-Down Menus --- contain a variety of useful commands, which you will discover as you go. All the commands in the Toolbar have equivalents on the pull-down menus.
6. Toolbar --- contains about ten of the most commonly used commands from the pull-down menus. The functions included depend on which window is "active". The Toolbar title bar gives the name of the active window. To change the active window move your cursor to the window you want to activate, and click with the left mouse button.
7. Function Keys --- located at the top of your keyboard, which you can use for navigating through DMS and ASSIST.

[See Appendix: A for more information]

Once you have invoked DMS you may invoke ASSIST. There are several ways of doing this:

1. Click the SAS/ASSIST icon  on the Toolbar, or
2. Select "Globals" from the pop-up menu (right mouse button) or pull-down menu bar and then select SAS/ASSIST, or
3. Type the word ASSIST in the command window at the top of the screen and press enter.

The SAS/ASSIST Main Menu will appear and look like:



The first icon to notice is the top left "Tutorial" icon. If you click on this icon you will find two useful items:



For new SAS users – covers the basic principles of the SAS System.



For navigating through ASSIST – covers the five ways of entering commands:

1. Pop-up menu (right mouse button)
2. Toolbar at top of screen
3. Command window (small window at top of screen)
4. Function keys (Appendix A)
5. Pull-down menus.

[In this tutorial we will always use either the pop-up menu or function keys to navigate through ASSIST.]

INDEX

The other icon to immediately explore is the Index icon. By clicking on any item listed in the Index, you can go directly to that item. Try this out:

1. Select "Index"
2. Scroll down until you find "REG procedure ..." and click on it
3. The "Regression Analysis" window will appear and if you enter the appropriate information, you may run a PROC REG analysis
4. Press the END key (F3) as often as necessary to return to the Primary Menu

Exercise 2 — Assign a LIBREF

SETUP

Perhaps the most important icon is the "SETUP" icon. Here you will assign "filerefs" and "librefs," to tell the SAS System where your flat files are located and where your SAS libraries are located.

[Under most operating systems, you cannot assign a fileref or a libref to a file or SAS library that doesn't exist, otherwise an error message will appear.]

1. Follow this path from the Primary Menu:

SETUP

SAS data libraries

List current libref

The "Libnames" window will appear with a list of the librefs currently in effect. This is not a necessary step but is helpful in verifying the next step.

2. Press the END key (F3) to return to the "SAS data libraries" window.
3. This time select **Assign a new libref**

As a practice exercise, fill in the fields on the "Assign a New Libref" windows as follows:



4. Type "outsas" as the Libref, this is a nickname associated with a SAS data library. For the name of the SAS data library type:

"A:\SUGI22\TUTOR\SASFILES" to specify that your SAS library is the \SUGI22\TUTOR\SASFILES directory of your A: drive.

5. Click on **OK**

In our example, under the Windows operating system, a window appears to tell you that the libref "OUTSAS" has been successfully assigned.

6. Select **List Current Librefs...** again. This time in the "Libnames" window you will see :
"OUTSAS V611 A:\"
7. To return to the Primary Menu press the END key (F3) as often as necessary.

Exercise 3 -- Entering Data Interactively

Once you have specified in "SETUP" where to read and write your files, you are ready to perform any number of tasks that ASSIST makes so easy to accomplish. One common task is entering data interactively into a SAS data set.

1. From the Primary Menu follow this path:



2. The "Enter data interactively" window appears. You can select either "Enter data one record at a time..." or "Enter data in Tabular form..."

For this example we will enter data one record at a time because we find it easier to verify that we have entered the data correctly.

3. The "Select New SAS Data Set to Create" window will appear. For this exercise fill in the fields as follows:

- **Data Set** Type "class" as the name of the data set.
- **Permanent** Click on this button and the "Libnames" window will appear with a list of available librefs. Since we earlier assigned the libref "OUTSAS", select it. Press ENTER.

4. The "Select a New Data Set to Create" window will reappear with the information you have entered. If everything is correct, click on **OK**

5. The "Define a New SAS Data Set" window will reappear. For the remaining steps use this codebook and sample data:

Codebook:

Variable	Type	Length	Label
NAME	\$	8	First Name
SEX	\$	1	F=Female M=Male
AGE	N	8	Age in Years
HEIGHT	N	8	Height in inches
WEIGHT	N	8	Weight in pounds

Data:

Name	Sex	Age	Height	Weight
Alice	F	13	56.5	84
Becky	F	13	65.3	98
Gail	F	14	64.3	90
Karen	F	12	56.3	77

Using the Tab key to move from field to field, enter the data as follows:

- a) In the NAME column, type the name of the new variable, as shown in the codebook.
- b) In the TYPE column, type \$ or a letter "c" to indicate that the variable is a character variable, i.e., a letter, a number, special character and symbol. Numeric variables, indicated with an "N", must be numbers and unlike character variables can be used in arithmetic calculations.
- c) In the LENGTH column, type the largest number of characters contained in the character variable. If the variable is numeric, you can specify a length of eight or leave the field blank (8 bytes is the default).
- d) In the LABEL column, type a description of the variable, as shown in the codebook.
6. Having finished defining your new SAS data set, press the END key (F3) and the new SAS data set will actually be created.

7. The "New" data entry screen will appear. You will enter your data into your new SAS data set in this screen. If you are entering data one record at the time, to add a new record click on :
Pop-up menu ► "Edit" ► "add new record"
8. When you have finished entering your data, press the End key (F3). At this point your SAS data set will contain all of the data you just entered.
9. To return to the Primary Menu, press the END key (F3) as often as necessary.

Exercise 4 — Exporting Data

Creating a flat file from a SAS data set is a task you will do less frequently, but might sometimes find necessary. You might want to do this, for instance, if you exchange data with a site that may not have SAS available.

[The flat file you create in this exercise will be used to create a SAS data set in our next exercise.]

1. Follow this path from the Primary menu:



2. The "Export data to flat file" window appears. For this exercise enter the following information:

Active data set: Click on this button, and a list of available SAS files will appear. Choose the active data set SASUSER.VENEER.

Export to file: - REQUIRED -

Type over - REQUIRED - with:
A:\SUGI22\TUTOR\DATA\VENEER.TXT
This will create the flat file "Veneer.txt" in the directory SUGI22\TUTOR\DATA of your A: drive.

Format output using:

The items under this heading determine how your flat file will be formatted:

- Click on "fixed columns" to place the data in fixed columns, or
- Click on "Separator" to format the data separated by tokens.
Token: < , > type the character you want to use inside this field.

Quotes:

The items under this heading specify whether you want the values in the external file separated by quotes.

- Click on either "None," "Single Quotes," or "Double Quotes."

Quote numeric values:

The items under this heading specify whether you want the numeric values quoted in the external file. Select "YES" even if your numeric values contain characters, commas or periods.

- Click on "YES," or "NO".

3. After you have finished making your selections in the "Export data to a flat file" screen, click on:
Pop-up menu ► "Locals" ► "Run"
4. A window will appear telling you that your file has been created and asking you if you want to browse the file. Click on "YES" and a screen will appear displaying the values in your flat file.
5. To return to the Primary Menu press the END key (F3) as often as necessary.

Exercise 5 — Importing Data

In the research environment, you will usually receive data in an existing "flat" file rather than having to enter data yourself. In this more common situation, you will want to "IMPORT" the file to convert it into a SAS data set.

1. Follow this path from the Primary Menu:



Import data from flat file

The "Import data" window will appear. For this exercise fill in the fields on this screen as follows:

2. **Data file to import:** - REQUIRED -

Click on - REQUIRED - and the "Select External File Name" window appears and you will see:

External file name:

For this exercise type in:
A:\SUGIZZTUTOR\DATA\VENEER.TXT,
the flat file we created in the last exercise.

Filename Click on the button in "Filename" to specify that you have entered the entire path of the flat file location.

Click on **OK** the "Import data" window will reappear.

3. **Output data set:** - REQUIRED -

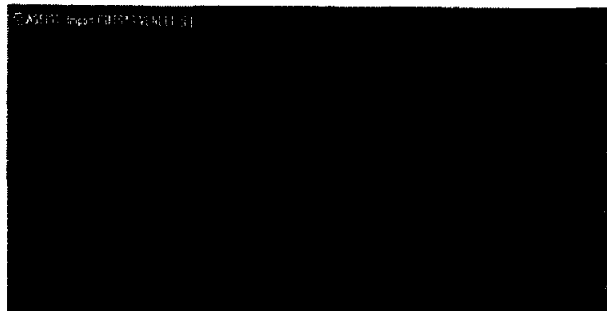
When you click on - REQUIRED - the "Specify Output Data Set" window will appear and you will see:

Data Set: For this exercise you may call your resulting SAS data set VENEER. Note that when created the system will add the "SSD" extension, so the SAS file will display in your directory as "VENEER.SSD."

Permanent. Click on the button to specify that you want to create a permanent SAS data set. The "Libnames" window will appear. For this exercise select the SAS library that we assigned in Exercise 1 with the libref "OUTSAS"

4. The "Specify Output" window will reappear so that you may verify the information you entered. If correct click on **OK**

5. The "Import Data" window will reappear so that you may verify all the information you entered. If correct click on **OK**
6. The "define fields" window will appear. We are NOT going to use this method of data definitions. To by-pass this window, click on **OK**
7. The "Import" window will appear and you will see:



To fill in the fields for this exercise use the following codebook:

Codebook:

Variable	Type	Columns	Label
BRAND	\$	1-5	Brand Name
WEAR	N	15-22	Durability

Use the tab key to move from field to field and the Page Down key to move to the next variable.

Name In the name field; type the name of the variable as specified in the codebook.

Start End Specify the column width of your variables in the "start" and "end" fields. Although you may leave these blanks, it is usually desirable to specify the columns where the variable is located. In this exercise, use the columns as specified in the codebook.

Type: You need to specify the type of variable by clicking on "Num" or "Char" for each variable.

Format: You may choose to enter a format for displaying the variable value. You may

type a "?" in the format field for a list of available formats.

Informat: You need to specify an informat to correctly read some variable values. You may type a "?" in the informat field for a list of available informats.

Label: You may enter a label for each variable. For this exercise use the labels as specified in the codebook.

8. When you have finished defining variables, click on:
the Pop-up menu ► Locals ► Create new data set...
9. The "Create new data set" window will appear. Click on "Verify" and "Report."
Click on **OK**
10. The "Statistics" window will appear with a report about the validity of data entered into the new SAS data set. Also there will be a message to "Press ENTER to remove Statistics". Press ENTER.
11. To return to the Primary Menu press the END key (F3) as often as necessary.

Exercise 6 -- Editing and Browsing Data

You can browse or edit a SAS data set using ASSIST.

1. Follow this path from the Primary Menu:



2. The "Edit Data" window will appear.

Tabular format

Click on the button in order to edit or browse multiple observations. ("Single Observation" will allow you to edit or browse one observation at a time.) The "Edit data in tabular format" window will appear. The name of the last data set in use will appear next to "Active data

set." You can select a different data set by clicking on "Active data set," to display a list of available data sets.

For this exercise, keep the SAS data set, VENEER that you created in exercise 5.

3. Click on **RUN**

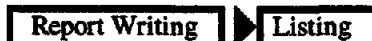
The data in your data set, in this exercise VENEER, will appear on the screen.

4. Click on the observation that you want to edit and use the Tab key to move between the variables within that observation. Once within a variable, you may have to move the cursor to the data that you want to change. Type the new data over the old data. If desired you may click on:
the pop-up menu ► "EDIT" to perform operations such as duplicating or deleting the observation currently selected.
5. To return to the Primary Menu press the END key (F3) as often as necessary.

Exercise 7 -- Producing and Printing a Listing Report

ASSIST allows you to produce many different types of reports such as a tabular, graphic, and listing reports. Tabular reports display data in a table format. You can insert means, sums, and percentages in tabular reports. Graphic reports display data in bar charts, pie charts, plots and maps. Listing reports display data for selected variables, showing one line for each observation in the data set.

1. Follow this path from the Primary Menu:



The "List a Data Set" window will appear. For this exercise enter the following information:

Active Data Set Choose from the available data sets, choose "SASUSER.FITNESS."

Variables to appear in report Select the desired variables, click on "Age," "Weight," and "Runtime."



2. Click on **OK**
3. Click on the pop-up menu **► "Locals" ►
"Run"**
Your listing report will appear in the
OUTPUT window.
4. To return to the "List Data Set" window,
press the END key (F3).

Now that you have created a listing report, you may send it to a printer for printing. From the "List a data Set" window follow this path:

5. Click on the pop-up menu **► **File** ►
Print Last Output..**
6. To return to the Primary Menu press the
END key (F3) as often as necessary.

Exercise 8 --- Saving Source Code

Why save the source code? There are several good reasons. First as discussed earlier, by saving your source code and studying it later, it's a great way to learn the SAS programming language. Secondly, source code documents your analytic strategy and enables you to replicate your results. And finally, if you are an experienced programmer you can use the source code as a template for further customization. Remember that ASSIST produces the source code, i.e. a SAS program, for every task you perform, and this code is documented.

Since every task you do in ASSIST generates source code, we could have selected any of our earlier tasks as an illustration. But for this exercise, we will look at a Data Analysis task.

1. Follow this path from the Primary Menu:



2. The "Summary Statistics" window will appear. For this exercise enter the following information:

Active Data Set: Click on the button and select: SASUSER.CRIME.

3. Click on **OK**
Variables Click on the button and
select: STATE, ROBBERY, and BUGLARY.

4. Click on **OK**
Output data set: click on the button. The
"Output Data Set" window will appear.

Enter: "TEST" as the "Data Set."

Click on **Temporary** and then **OK**

5. Click on "Sum" "Mean" and "Standard
Deviation," to choose some statistics for
this exercise.
6. Click on the pop-up menu **► "Locals" ►
"Run"**
The Output window will appear with your
results.
7. Press the END key (F3) to return to the
"Summary Statistics" window.

Now that you have performed an ASSIST task, you may look at the code generated by ASSIST; save the code; further use it by modifying it, rerunning it, or making a copy of it, and so forth.

Viewing the Code

To look at the code:

1. From the "Summary Statistics" window,
click on the pop-up menu **► "Locals"**
2. Click on "View Source." As you see,
ASSIST has written documented SAS
program, based on the selections you
made.
3. Press the END key (F3) to return to the
"Summary Statistics" window.

Saving the Code

Besides viewing the program, you may also save it as either a flat file or a SAS catalog. A SAS catalog is a particular kind of SAS file, when saving your source code you need to consider whether you want to use that code

within ASSIST or outside of ASSIST. If, you want to reuse your SAS code inside ASSIST, save it as a SAS catalog, otherwise save it as a flat file.

To Save the Code as a Flat File

From the "Summary Statistics" window, follow these steps:

1. From the pop-up menu, follow this path:



2. The "View SAS program" window appears. From the pop-up menu, follow this path:



3. The "save As" window appears. For this exercise enter the following information:

Save as filename: Type in a name for the program file. For this exercise "prog1"

Drive: Select A:

Save file as type: Select <All files>

Save AS Click on when you have finished filling in the information.

4. A message will appear at the bottom of the screen confirming that your SAS program has been saved. The "View" window will reappear.
5. Press the END key (F3) to return to the "Summary Statistics" window.

To Save Code as a SAS Catalog

From the "Summary Statistics" window, follow these directions.

1. From the pop-up menu follow this path:



2. The "Save as" window will appear, for this exercise fill the fields as follows

Name: Enter a name for the source file, enter "prog2"

Description: Enter a description of the program, enter "Crime Data Analysis Program".

Catalog: SASUSER.PRIVATE

By default your source file will be saved in the catalog named "SASUSER.PRIVATE." To save the program in a different SAS catalog, you can either type over SASUSER.PRIVATE with a new catalog name or you can type a "?" in the field to see a list of available catalogs.

Now that you have saved your source code (program) as a SAS catalog, ASSIST provides some handy ways of using that code:

1. From the "Summary Statistics" window, select "Locals" from the pop-up menu.
2. Select "Results Manager".
3. The "Result Manager" window will appear with a list of your saved programs. Type a "?" in from of the program name field and these options will be displayed:

B	Browse
C	Copy
D	Delete
E	Edit
F	File
L	Batch Execution
P	Print
R	Rename
V	Verify delete
X	Execute

Now you may perform any of the tasks specified – when you are finished you may return to the Primary Menu by repeatedly pressing the END key (F3).

Exercise 9 -- Exit ASSIST

To exit ASSIST, always return to the Primary Menu. You can get to the Primary Menu from any lower-level menu by clicking on the END key (F3) as often as necessary.

When you are in the Primary menu, click on the "EXIT" icon. You will then be in your SAS session and you may do more work or terminate your session.

Conclusion

After a number of years of teaching introductory SAS/ASSIST, we find ASSIST to be easy to use, flexible, and a valuable aid in using the SAS System. SAS/ASSIST is also frankly fun to use. Because ASSIST produces source code for every task you perform, it enables you to document your analytical strategy, reproduce your results, and study that code in order to learn the SAS programming language. We believe that the inclusion of SAS/ASSIST in any introductory class should be seriously considered.

Author's Addresses

Sally Muller
Academic Technology and Networks
310 Wilson Library
CB#3460
University of North Carolina
Chapel Hill, NC 27599-3460
Phone: (919) 966 1533
Fax: (919) 962 2654
EMAIL: Sally_Muller@unc.edu

Arturo Barrios
Academic Technology and Networks
310 Wilson Library
CB#3460
University of North Carolina
Chapel Hill, NC 27599-3460
Phone: (919) 966 1533
Fax: (919) 962 2654
EMAIL: Arturo_Barrios@unc.edu

Acknowledgements

Kenneth A. Hardy, of the Institute for Research in Social Sciences at the University of North Carolina, who reviewed this paper and offered many valuable suggestions. Thanks also are extended to the members of the SAS Institute Technical Support Department who answered our questions along the way.

References

Getting Started with the SAS System Using SAS/ASSIST Software. Version 6, Second Edition, SAS Institute Inc., 1996.

SAS/ASSIST Software: Changes and Enhancements. Version 6, First Edition, SAS Institute Inc., 1995.

Doing More with SAS/ASSIST Software. Version 6, First Edition, SAS Institute Inc. 1992.

Spicer, Jeanne (1995) "A Brief Introduction to SAS and SAS/ASSIST."

SAS Language & Procedures: Introduction, Version 6, Cary, N.C.: SAS Institute Inc., 1990.

SAS Video tutorial, Making your move to the SAS/ASSIST Software, Cary, N.C.: SAS Institute, 1991.

Appendix A: Function Keys

At the top of your keyboard are keys that begin with a letter "F", which you can use to navigate through SAS/ASSIST. They are called "Function Keys." Function keys allow you to perform actions quickly. The function keys change depending on where you are in ASSIST. There are two ways of viewing function key definitions.

- Type "keys" on the command line at any point
- use the function key F9 to call up a list of function keys

Useful navigation commands and their function keys include:

HELP	Accesses the help menu	(F1)
END	Closes the window	(F3)
FORWARD	Shows next window	(Shift F10)
BACKWARD	Shows previous screen	(Shift F9)
RIGHT	Shows window to right of current window	(Shift F8)
LEFT	Shows window to left of current window	(Shift F7)
PMENU	Toggles between the menu bar and command line	(F11)

You may assign a command function key if so desired. For example you may wish to assign a function key for "CLEAR."

Appendix B: Basic SAS Concepts

Thanks to Jeanne Spicer, from whose paper we heavily borrowed the following information.

SAS Program

A SAS program is a sequence of steps that you submit for execution. A SAS program can consist of two kind of steps:

DATA step – typically to create a SAS data set, but also can be use to create reports.

PROC step – to process SAS data sets (generate reports and graphs, edit data, sort data, and so on) and to create SAS data sets.

A SAS program may contain many numbers of DATA steps and PROC steps. These PROCedures are typically used on data that is in SAS format called SAS data set.

Basic SAS Programming Statements

All SAS programs follow the same basic rules:

1. Each SAS statement is ended or separated by a semicolon (;).
2. All variables and data step names should be no more than 8 characters in length and they must start with a letter. Characters /, &, and ; are not allowed.
3. The SAS language is not case sensitive. Lower or upper case type or a combination may be used.
4. SAS statements can span multiple lines.
5. Multiple statements can appear on one line, as long as they are separated by a semicolon.

More on the Structure of a SAS Program

1. Comment statement – Two forms are valid in SAS:
 - a) An asterisk in column 1 starts a comment line and a semicolon ends it.
* put your comment here ;
 - b) A slash-asterisk (/*) tells SAS to ignore everything until it finds asterisk-slash (*).
/* section of code that you don't

want to execute this time but
you might want later */

2. Options statements – Many system options may be set here:

```
Options compress = yes /* compress
SAS data file */
replace ; /* permit overwrite
of data sets */
```

3. Filename statements – Tells the SAS system where your flat file is located. Use the "Open Folder" icon on the toolbar to bring up SAS programs into the Program editor window. Also use the "open Folder" icon to examine a flat file.
4. Libname statement –
 - a) SAS Library – A collection of one or more SAS files, including SAS data sets, which the SAS system recognizes. Each SAS file is a member of a library.
 - b) SAS libref – A SAS libref is a temporary and arbitrary name pointing to a SAS library. The name of a permanent SAS file consists of two words separated by a period. The libref is the first word and points to the library; the second word is the name of the SAS file in the library. Therefore, in ASSIST, if you assign a SAS file the name "class" and your libref is "myfiles," to access the file you will specify "myfiles.class."
5. Data step statement – The main function of the DATA step is to construct data sets. Data may be included in your program or reside as a separate file on the network.

Getting data into a form that SAS can process is probably the most difficult aspect of coding. It may be helpful to think of SAS data set as a rectangular table, like a spreadsheet: data values in a row comprise an "observation," and those in a column comprise a "variable." Once the data is in a SAS data set it may be accessed directly by any PROC step or DATA step.

Data can be typed into a SAS data file directly, but usually you will be reading data in from one of 3 locations:

1. Data in a job stream.
Small amounts of data can be entered directly into SAS source code that includes

the data step. In this case the following SAS data steps statements need to be used:
Example SAS code:

```
data new;
input x y $ z;
datalines;
1 M 3
2 F 9
3 M
; /* don't forget the semicolon */
run;
```

The datelines statement alerts SAS that data will follow. The data set "new" is now available for processing in subsequent SAS procedures.

2. Data in flat files

Larger amounts of data are storage in separate files. You can create your own data files using ASSIST or a text editor, or you may want to access data prepared by someone else that already exists as a file. Prior to your data step, use the filename statement to specify the location of your data.

Example SAS code:

```
filename ex 'a:\sugi22\begin\tutor\myfiles.dat';
data in;
infile ex;
input x y z ;
run;
```

3. Existing SAS data sets (also referred to as SAS system files or SAS data files)
Data that is already in SAS data file can be reused without having to specify the names of the input variables. You will still need to specify the library containing the data with a libname statement. However, no infile statement is required and the complicated "input" statement is replaced with a "set" statement.

Example SAS code:

```
libname ex2 'a:';
data new;
set ex2.testfile (obs=5);
run;
```

Note the "obs=" option in parentheses in the "set" statement. In this program the temporary data set "new", is created consisting of only 5 observations from the

original testfile. All variables, labels, and formats, that were stored on file "testfile", are available. This data subset may be manipulated or procedures may be run on it by referring to data step "new". The original SAS system file "testfile" remains untouched.

6. Input statements – You can combine the following three modes of input in a single input statement:
 - a) Specifying column numbers in an input statement is called "column mode" input.
 - b) Specifying just the variable names in an input statement is called "list" or "free-form" mode input.
 - c) Specifying formats and pointers in an input statement is called "formatted" mode input.

Each mode of input has its own advantages and disadvantages.

7. Proc statement -- The following are some commonly used procedures. Words in bold type are "keywords" required by the PROC that you copy right out of the manual. The words in *italic* are supplied by you. Try typing these statements (a-d) in the Program editor window and submitting them with the F3 key. Be sure to examine both the log and output windows.

- a) To display the header information from a SAS data set:
proc contents data=sasuser.build;
run;
- b) Printing - print values in the two variables pop80 & over65 for the first 5 observations only.
proc print data=sasuser.class
(obs=5);
var pop80 over65;
run;
- c) Frequency tables - create frequency distribution of income level in each region and check for bias.
proc freq data=sasuser.crime;
tables burglary * state /chisq;
run;
- d) Descriptive statistics - measures of central tendency, variation & shape.
proc means data=sasuser.fitness;
var weight runtime /skewness
kurtosis;
run;

- e) Formats - create a library of definitions you can use to write out the values of variables in different ways.

```
proc format;
value hlow low-49='low'
          50-high='high';
value $gender 'F'='Female'
              'M'='Male';

run;
```

Appendix C: Producing a Listing report of a Subset of Data

Steps to produce a report from a subset of the data set:

1. From the Primary Menu follow this path: "Report Writing" ► "Listing"
2. The "List a data set" window appears. Select "Subset data".
3. Select "Where clause."
4. The "Build a where clause" window allows you to select data for the new report.
 - a) If you know how to describe the data for the new report, click on "Edit the where clause" window and enter the variable names in addition to the correct comparison operators and values.
 - b) If you don't know how to specify the "Where clause" desired, you can build the where clause by selecting appropriate variables names, comparison operators, and values from among those listed in the "Build a where clause" window. To build a where clause follow these steps:
 - Select "variable" and a list of variables in the data set will appear. Select a variable, hit ENTER, and the "Build a where clause" window reappears.
 - Select "Comparison operator" from the "Build a where clause" window.
 - Select an appropriate comparison operator from those listed in the window and the "Build a where clause" window reappears.
 - Select "Constant . . ." and enter the constant. The constant will be the standard of comparison

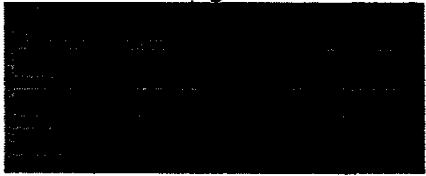
determining which variables will be included in the new data set. For example 3 is the constant in the where clause "age=3". The equal sign is the comparison operator and "age" is the variable.

- Select OK. You can change the where clause by selecting "Edit the where clause."
6. Select "locals" from the menu bar and select RUN. The output window will show the report.
 7. Use the END function key (F3) to return to the "List a data set" window.

Appendix D: Merging SAS Data Sets

You can use "combine" in the Data Management menu to concatenate, interleave, merge, or match-merge your data. Match-merge allows you to perform a SAS match merge or a Cartesian product match merge. The Cartesian product match merge uses PROC SQL statements to merge the data sets or views. When you use the Cartesian product match merge, your data sets don't have to be sorted by the variable(s) by which you merge. Another advantage of the Cartesian method is that you can merge by unique variable(s).

In this example we will use "Combine" on the Data management menu to perform a Cartesian product match merge. You can use the "Combine" for any type of SAS data set. Follow these steps to perform a Cartesian product match merge:

1. From the Primary Menu, select Data MGMT.
 2. Select "Combine"
 3. Select "Match Merge"
- 
4. Select "Second Method."
 5. Select "First Data Set" to display a list of available data sets and then choose the data set that you want to merge. Follow the same procedure for "Second Data Set."

6. Select "Output Data." The Output data set or view window appears.
 - a) In the "data set/view" window, type the name of the new file where the merged data will be stored. Please note that the file name can only be eight characters long
 - b) Select "Temporary" or "Permanent"
 - c) Select OK to return to the "Match merge data set" window.
7. Select "Type of variable used to match merge the data sets." The "Select a merge method" window appears.
8. Select "Combine by variables unique to each data set," assuming that you want to combine variables of the same data type in your two data sets, but these variables has different names (e.g. "STUDID" in one data set and "ID" on the other data set). The "match merge by different variables" window appears.
 - a) Select the variable from the first data set (e.g. "STUDID")
 - b) Select the variable from the second data set (e.g. "ID")
 - c) Select OK to return to the "Match merge SAS data sets" window.
9. Select "Select existing variables."
10. Select "Unique existing variables."
11. The "Select existing variables" window appears.
12. Select all the variables that you want to include in the new merged file.
13. Select OK to return to the "Match merge SAS data sets" window.
14. Select "Locals on the menu bar then RUN.
15. Click OK and then ENTER.
16. The "Combined data" window appears.
17. To return to the Primary menu, from the menu bar select File and then End.