

CHAPTER

1

GETTING STARTED

LEARNING OBJECTIVES

- To be able to use the SAS software program in a Windows environment
- To understand basic information about getting data into SAS and running a SAS program
- To be able to run a simple SAS program

2 Getting Started

Research involves the collection and analysis of data. The SAS system is a powerful software program designed to give researchers a wide variety of both data management and data analysis capabilities. Although SAS has millions of users worldwide, it is not the simplest program to learn. With that in mind, we've created this book to provide you with a straightforward approach to learning SAS that can help you surmount the learning curve and successfully use SAS for data analysis.

Two main concepts are involved in learning SAS: first, how to get data into SAS and ready for analysis, and second, how to perform the desired data analysis. This chapter introduces you to the SAS system's use in the Microsoft Windows environment and provides examples of using SAS to analyze data.

USING SAS IN A WINDOWS ENVIRONMENT

SAS runs on a number of computer platforms (operating systems), including mainframes and personal computers whose operating systems are UNIX, Mac OS X, Linux, and Windows. This book is based on SAS's use in the Windows environment, although most of what is covered is the same on any computer. We assume that you have already installed SAS on your computer. Most differences across computers have to do with file references. Before discussing the SAS program, we'll review some of the basic file-handling features of Windows.

Creating a Folder for Storing Your SAS Files

Because there are several versions of Windows currently in use, we present general guidelines that should work in any one of them. To follow our examples, we recommend that you store the data and SAS files provided with this book in a folder named *SASDATA*. In fact, the examples in this text will assume that all example files used in this book are at the root of your hard drive, in a folder whose full path is `C:\SASDATA`.

In your own research, you may choose to store your SAS files in a folder with any name that makes sense to you, such as `C:\RESEARCH`. You can also create subfolders for each analysis—such as `C:\RESEARCH\SMITH` or `C:\RESEARCH\JONES`.

To copy the example files for this book to the `C:\SASDATA` folder on your own computer, do the following:

1. Download the example files from the following Website:
<http://www.alanelliott.com/sas>
2. Follow the installation instructions provided on the Web site. Doing so creates the `C:\SASDATA` folder on your computer and copies the example files to that folder. (The Web site may also include updates concerning the information in this book.)

The examples in the book are designed to use data in the folder `C:\SASDATA`—in other words, in a folder on your computer’s hard drive. You can also put the example files on any rewritable medium such as a flash drive—just remember to adjust the file names and file paths given in this book accordingly if you store your files in a location other than `C:\SASDATA`.

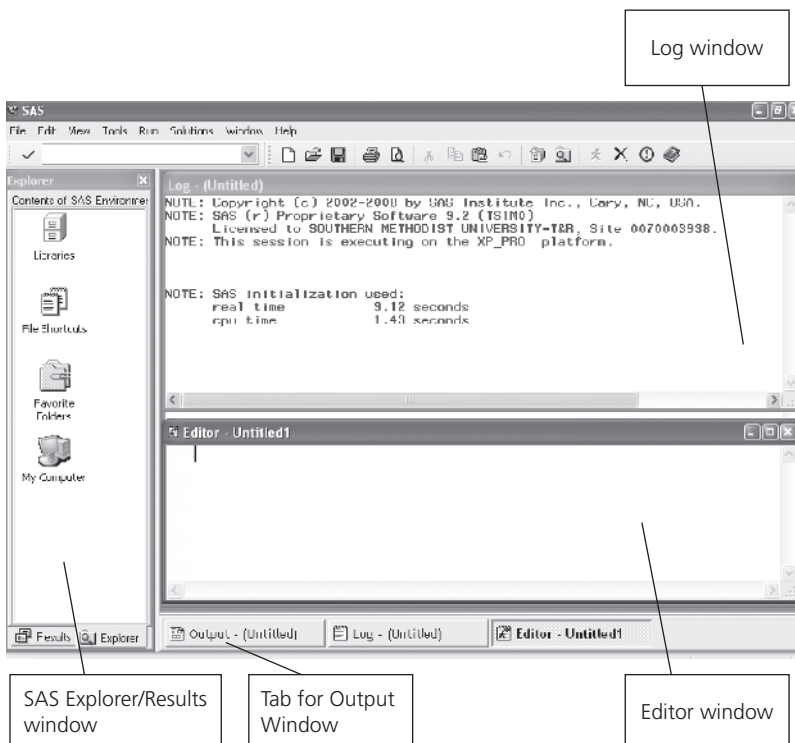
Beginning the SAS Program

There are at least two ways to start the SAS program. If the SAS icon appears on your Windows desktop, simply double-click it to launch SAS. If the SAS icon is not on your desktop, go to the Start button and select Start ⇒ Programs ⇒ SAS (English) to launch SAS. Henceforth we will refer to this simply as the SAS icon.

Understanding the SAS Windows Interface

Once you begin the SAS program, you will see a screen similar to that shown in Figure 1.1. (The SAS program appearance may be slightly different depending on which versions of Windows and SAS you’re using.)

FIGURE 1.1. *Initial SAS screen*



4 Getting Started

The normal opening SAS screen is divided into three visible windows. The top right window is the **Log** window, and at the bottom right is the **Editor**. The third window, which appears as a vertical element on the left, is called the **SAS Explorer/Results** window. There are other SAS windows that are not currently visible. These include the **Output** and **Graph** windows. (A tab for the Output window does appear on the initial screen.) To open a SAS window that is not currently visible, click its tab at the bottom of the screen. Briefly, here are the purposes of these windows.

Editor: Also called the Enhanced Editor or Windows Programming Editor (WPGM), this is the area where you write SAS code. It is like a simple word processor. When you open a previously saved SAS program, its contents will appear in this window. SAS code is stored in plain ASCII text, so files saved in the ASCII format from any other editor or word processor may be easily opened in this editor. You can also copy (or cut) text from another editor or word processor and paste it into the Editor window.

Log: When you run a SAS program, a report detailing how (and if) the program ran appears in the Log window. Typically, when you run a SAS program, you first look at the contents of the Log window to see if any errors in the program were reported. The Log highlights errors in red. You should also look for warnings and other notes in the Log window which tell you that some part of your program may not have run correctly.

SAS Explorer/Results: This window appears at the left of the screen and contains two tabs (shown at the bottom of the window): The Results tab displays a tree-like listing of your output, making it easy to scroll quickly to any place in your output listing. The Explorer tab (currently shown in Figure 1.1) displays the available SAS libraries, where SAS data files are stored. A SAS library is a nickname to an actual physical location on disk, such as `C:\SASDATA`. This will be described in detail in Chapter 3.

Output: Once you run a SAS program that creates analysis output, the Output window is automatically displayed. This window contains the results (also called the listing) of your SAS job. If it does not appear, click the Output tab to display this window.

Graph: If your SAS program creates graphic output, SAS will display a Graph window containing that output. Usually such output is displayed by default. If it does not appear, click the Graph tab (not shown in Figure 1.1) to display this window.

Do not close any of the windows that make up the SAS interface. Move from window to window by clicking on the tabs (which are analogous to the tabs on the Windows taskbar) or by pressing an appropriate function key. If you close one of the windows, its tab at the bottom of the SAS screen will disappear and you will need to go to the View pull-down menu and select the appropriate window name to redisplay the element that you closed.

YOUR FIRST SAS ANALYSIS

Now that you have installed SAS and copied the SAS example data files to your computer, it's time to jump in and perform a quick analysis to see how the program works. (You have downloaded the example files, haven't you? If not, go back to the "Creating a Folder for Storing Your SAS Files" section and do so.) The following steps show you how to open a SAS program file, analyze the data in the file, and create statistical output.

Our first example is a quick overview of how SAS works. You should not be concerned if you don't know much about the information in the SAS program file at this point. The remainder of the book teaches you how to create and run SAS programs.

HANDS-ON EXAMPLE

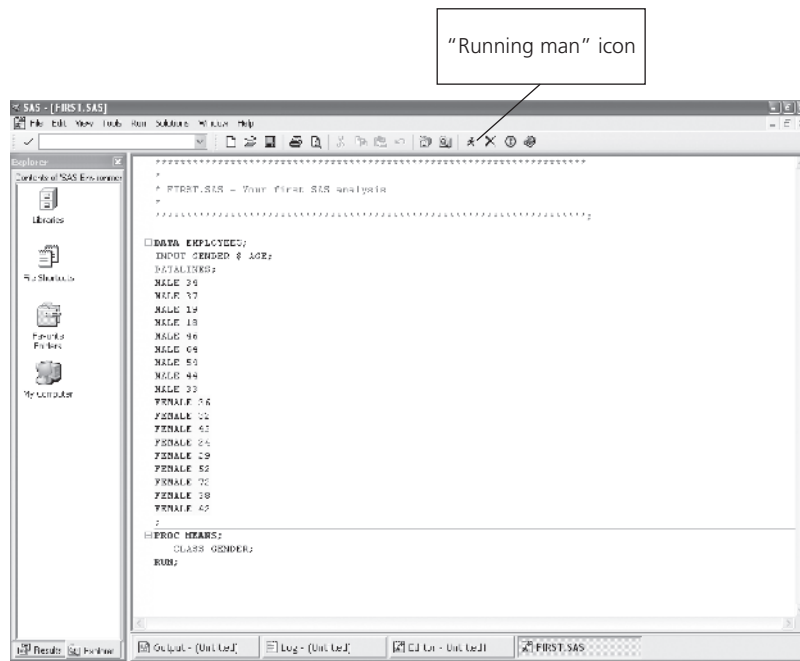
In this example you'll run your first SAS analysis.

1. Launch SAS by double-clicking the SAS icon on your desktop or by selecting Start ⇒ Programs ⇒ SAS. (To put the SAS program icon on your desktop, consult your Windows documentation.) SAS initially appears as shown in Figure 1.1. (If a tutorial or some other initial dialog box appears, dismiss it.)
2. Open a program file. This example uses a file named `FIRST.SAS`. To open this file, first make sure that your active window is the Editor window. (Click anywhere in the Editor window to make it the active window.) On the menu bar, select File ⇒ Open Program to open the file `C:\SASDATA\FIRST.SAS`. You may need to navigate to the `C:\SASDATA` folder to open this file.

(Continued)

3. Examine the opened file, the contents of which appear in the Editor window. Maximize the Editor window so you can see more of the program code. Figure 1.2 shows the Editor window maximized. We'll learn more in later chapters about what these lines of code mean.

FIGURE 1.2. *The FIRST.SAS file opened*



4. Run the SAS job. On the menu bar, select Run ⇒ Submit, click the "running man" icon on the toolbar, or press F8. The SAS instructions in the file are carried out (or, in computer terminology, the commands are executed). The SAS program creates analysis output that automatically appears in the Output window.
5. View the output. If the job runs without errors (which it should in this case), the Output window is displayed, showing the analysis output created by the program. The Output window is shown maximized in Figure 1.3.

FIGURE 1.3. Output for FIRST.SAS

GENDER	N Obs	N	Mean	Std Dev	Minimum	Maximum
FEMALE	9	9	40.0000000	14.2954927	24.0000000	72.0000000
MALE	9	9	30.7777770	15.1226467	10.0000000	64.0000000

- a. The analysis requested in this SAS program is a task to calculate basic statistics (PROC MEANS) for each group (GENDER) in the data set.
 - b. Notice the Results window on the left in Figure 1.3—it contains a single item called **Means: The SAS System** (the full name is truncated in the image). As you perform more analyses in SAS, this list will grow. You can quickly display the results of any previous analysis by clicking the item in the list.
6. Print the output. Make sure the Output window is active by pressing F7 or clicking anywhere in that window, then print the contents of the Output window by clicking the printer icon on the toolbar or by selecting File ⇒ Print on the menu bar.
 7. Save the output. To save the results shown in the Output window, on the menu bar select File ⇒ Save As and enter an appropriate name for the output file, such as FIRST OUTPUT. The file will be saved as FIRST OUTPUT .LST. (The .LST extension is automatically added to the file name.) (You must be in the Output window for the file to be saved properly.)

(Continued)

8. Save the SAS program (instruction file) that created the output by going to the Editor window and clicking the tab named `FIRST.SAS`, shown at the bottom right of Figure 1.3. With the program on the screen, select `File ⇒ Save As` and enter a name such as `MYFIRST`. This saves the file as `MYFIRST.SAS`. (The `.SAS` extension is automatically added to the file name.)
9. Clear the output from the SAS windows to get ready for the next example. In the Output window, right-click and select `Edit ⇒ Clear All` from the pop-up menu. Do the same for the Log window. In the Editor window, right-click and select `Clear All`. (We'll learn a shortcut to this process later in this chapter.)

The Save As command saves the contents of whatever window is currently active. Thus, if you are in the Output window, it saves the output as an `*.lst` (listing) file; if you are in the Log window, it saves the contents of that window as an `*.log` file; and if you are in the Editor window it saves the code as an `*.sas` (SAS program code) file.

That's it! You've run your first SAS job. But wait, there's more. Now that you have one SAS job under your belt, it's time to try another one. This time, you'll make a small change to the program before you run it.

Before running each Hands-on Example, clear the Output, Log, and Editor windows so you will not mix up data from a previous example. To clear each window, use the right-click technique previously described, or go to the window and on the menu bar select `Edit ⇒ Clear All`. In the section "Using SAS Function Keys" in this chapter we will create a way to shorten this process.

HANDS-ON EXAMPLE

In this example you will make a small change in the program and see how that change alters the output.

1. In the Editor window, open the file `SECOND.SAS`. (It should be in your `C:\SASDATA` folder.)
2. Under the comment line that reads


```
* PUT YOUR TITLE ON THE NEXT LINE;
```

 type this new line:


```
TITLE "HELLO WORLD, MY NAME IS your name.";
```

 Enter your own name instead of *your name*. Make sure you include the quotation marks and end the statement with a semicolon (;).
3. Run the SAS program, examine the output, and print the results.

HOW SAS WORKS

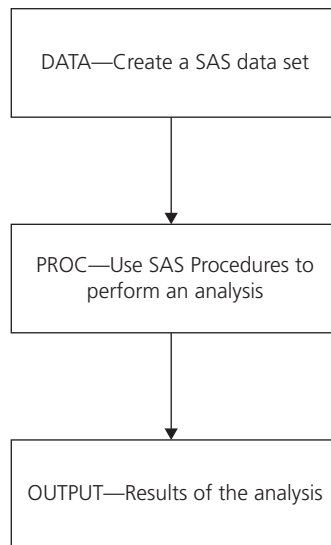
Look at the `SECOND.SAS` program file to see how SAS works. It is really very simple. The lines in the file are like the items on a grocery list or a “to do” list. You create a list of things you want SAS to do and when you submit the job, SAS does its best to carry out your instructions. The basic steps are:

1. Enter data into SAS. (This is called the `DATA` step.)
2. Tell SAS what analysis to perform. (This is called `PROCS`, for procedures.)
3. Observe the output. (This is called `OUTPUT`.)

Figure 1.4 illustrates this process.

This book follows the process shown in Figure 1.4. It contains a number of sample SAS programs that show you how to get data into SAS, analyze the data using SAS procedures, and read the output.

To end the SAS program, select `File ⇒ Exit` on the menu bar. Make sure you’ve saved all files you wish to save before ending the program. (The program will prompt you to save files if you have not previously done so.)

FIGURE 1.4. *How SAS works*

TIPS AND TRICKS FOR RUNNING SAS PROGRAMS

This section contains tips and tricks that can help you if something in your program goes wrong. Now that you have seen two sample SAS jobs, you should have an idea of how SAS programs are constructed. This section provides general rules for writing SAS programs.

Within a SAS program, each statement begins with an identifying keyword (DATA, PROC, INPUT, CARDS, RUN, etc.) and ends with a semicolon (;). For example:

```
DATA TEMP;  
PROC PRINT DATA=TEMP;
```

SAS statements are free-format—they do not have to begin in any particular column. Thus,

- Statements can begin anywhere and end anywhere.
- **One statement can continue over several lines** with a semicolon at the end of the last line signaling the end of the statement.

- **Several statements may be on the same line**, with each statement terminated with a semicolon.
- **Blanks**, as many as you want but at least one, may be used to separate the components (words) in a SAS program statement.
- Case (lower- and upper-) *doesn't* matter in most SAS statements.
- **Case does matter in data and quoted information.** When you enter character data values (such as M or m for “male”), case will matter when you are sorting or selecting the data. Also, the contents of a title or footnote are placed in quotation marks and are printed in the output using the uppercase and lowercase characters you indicate in the program code.
- **The most common error in SAS programming is a misplaced (or missing) semicolon.** Pay careful attention to the semicolons at the end of the statements; they tell SAS that the statement is completed.
- **A second common error is a missing RUN statement.** You *must* end every program with a RUN statement (in the Windows version of SAS). If you do not end the program with a RUN statement, your program may not terminate normally, and your last statement may not be completed.
- **A third common error in a SAS program is the presence of unmatched quotation marks.** If you enter a title without matching quotation marks, you will see an error message in your Log file such as “The TITLE statement is ambiguous due to . . . unquoted text.” This error can cause subsequent statements in your code to be misinterpreted by SAS.
- **A search for errors in a program log should be from the top down.** It is inevitable that you will write programs that contain errors in syntax. When you get lots of errors in a SAS run, always fix the errors from the top down in your program code, because often the early errors lead to confusion that results in the later errors.
- **If program errors cause problems that result in SAS's “freezing up” or not completing the steps in your program**, you can abort SAS's processes by pressing Ctrl+Break and selecting the “Cancel Submitted Statements” option from the dialog box that appears.
- **If you cannot resolve a problem within SAS**, *save your files*, restart SAS, and reopen the files. Try to find the code that caused the problem, and then re-run your program.
- **The structure of your SAS programs should be easy to read.** Notice how the example programs in this book are structured—how some statements are indented

12 Getting Started

for easy reading and how the code contains comments. Pay attention to the formatting of future examples. These commonly accepted (and recommended) program formatting techniques can help you or your colleagues in the future when you go back to read code that was written months or years before.

Now that you've experienced running simple programs in SAS, you will want to know more about the syntax and conventions for entering and running a SAS program. Following is a brief introduction to some of the SAS program requirements you need to know.

Using the SAS Enhanced Editor

You may have noticed that different parts of the coding in the `FIRST.SAS` file appear in various colors in the SAS Enhanced Editor. This editor is designed to help you write your SAS code and to prevent many potential errors. The color coding is designed to help you clearly see step-boundaries (major SAS commands), keywords, column and table names, and constants.

For example, in a SAS program:

Green is used for comments.

Dark Blue is used for the keyword in major SAS commands.

Blue is used for keywords that have special meaning as SAS commands.

A yellow highlight is used for data.

A boundary line separates each step.

If you make a mistake in coding your SAS job, the appropriate colors will not appear. In fact, statements that SAS does not understand typically appear in red, which helps you locate potential problems with your code.

Using SAS Function Keys

You can use keyboard function keys to move from window to window or to execute certain tasks in SAS. Some people prefer the use of function keys (rather than mouse clicks) as a quicker way of selecting program options. Table 1.1 lists some of the SAS function keys you can use throughout this book.

You can also customize these keys. For example, to display the full set of function key commands, on the menu bar select Tools ⇒ Options ⇒ Keys.

TABLE 1.1. SAS Function Keys

Function Key	SAS command	Result
F2	RESHOW	Reshows window interrupted by system command
F3	END; /*GSUBMIT BUFFER=DEFAULT*/	Submits SAS statements in clipboard
F4	RECALL	Recalls current SAS code to editor
F5	PROGRAM (PGM)	Displays Editor window
F6	LOG	Displays Log window
F7	OUTPUT	Displays Output window
F8	ZOOM OFF; SUBMIT	Submits (runs) the current SAS program
F9	KEYS	Displays Keys window
F10	Not defined	
F11	COMMAND BAR	Moves cursor to command bar
F12	Not defined	

HANDS-ON EXAMPLE

In earlier examples you cleared the information from the Output and Log windows manually. In this example you will create a new F12 function key command that will clear these windows and return you to the Editor window. The F12 key you're about to define uses three SAS commands: `CLEAR OUTPUT`, `CLEAR LOG`, and `WPGM`. The `CLEAR` command clears the contents of the Output and Log windows, and the `WPGM` command returns you to the Windows Programming Editor.

Follow these instructions to create a new F12 function key definition.

1. On the menu bar select Tools ⇒ Options ⇒ Keys.
2. Next to the blank F12 option, enter `CLEAR OUTPUT; CLEAR LOG; WPGM`. (Do not include the period that ends the previous sentence.) Press the Enter key to lock in the new command.
3. Exit the Keys window and try out this function key by re-running one of the previous examples. With the output displayed on the screen, press F12. The output will be cleared, the log file information will be cleared, and the Editor window will be displayed, still containing the current program code. Thus, this command allows you to quickly clear the Log and Output windows without deleting the program code. This new command will be used in some of our future examples.

Use the F12 SAS function key you created to clear the contents of the Output and Log windows between examples.

Using the SAS Menus

The SAS pull-down menu items differ depending on which window is currently active (Editor, Output, Log). To see the options change, click each window to make it active and notice the changes to the menu choices at the top of the main SAS window.

The SAS window you will use most is that of the Editor, because that is where you will write and edit SAS code. Therefore, we'll spend a little time describing the menus for the Editor window.

File: Used for opening and saving files and for printing.

Edit: Used to copy, cut, and paste text, as well as to find and replace text.

View: Allows you to go back and forth between viewing the Editor window, the Log window, and the Output window.

Tools: Allows you to open programs for graphics and text editing, along with other options available to customize the program to your preferences.

Run: Allows you to run (submit) a SAS program; also provides access to remote SAS options.

Solutions: Contains options to advanced procedures not discussed in this text.

Window: As in most Windows programs, allows you to choose display options for opened windows such as tile and cascade; also allows you to select (make active) a particular window, such as the Log or Output window.

Help: Contains options for the SAS help system as well as online documentation. (See the section “Getting SAS Help on the Web” later in this chapter.)

Understanding Common File Extensions

Another piece of information you need to be aware of when learning and using SAS concerns the type of files used and created by SAS or mentioned in this book. Like most Windows files, they have specific file name extensions that indicate what application is associated with them (e.g., the way a *.doc extension indicates a Microsoft Word file).

SAS code file (*filename.sas*): This is an ASCII text file and may be edited using the SAS Editor, Notepad, or any text editor that can read an ASCII file.

SAS log file (*filename.log*): This ASCII text file contains information such as errors, warnings, and data set details for a submitted SAS program.

SAS listing file (*filename.lst*): This ASCII file contains the output (listing) of a SAS job.

SAS data file (*filename.sas7bdat*): This file contains a SAS data set, including variable names, labels, and the results of calculations you may have performed in a DATA step. You cannot open or read this file except in SAS or in a few other programs that can read SAS data files.

Raw data files (*filename.dat* or *filename.txt* or *filename.csv*): These ASCII (text) files contain raw data that can be imported into SAS or edited in an editor such as Notepad.

Excel file (*filename.xls*): The data in a Microsoft Excel file (when properly formatted into a table of columnar data) can be imported into SAS using the File ⇒ Import menu selection. Other data files such as those produced by Microsoft

16 Getting Started

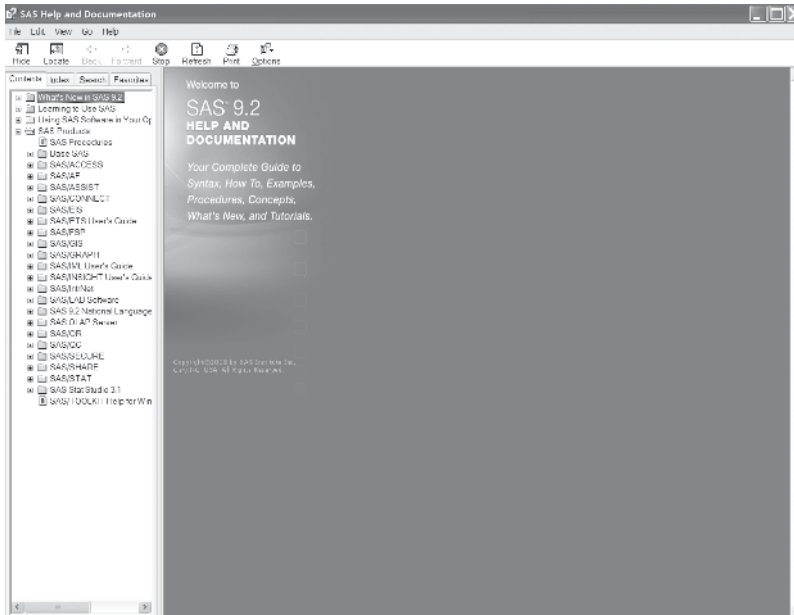
Access (*.mdb) and Lotus 1-2-3 (*.wks) can be imported into SAS in a similar way.

Getting SAS Help on the Web

SAS help is available from the SAS Help pull-down menu (Help ⇒ SAS Help and Documentation). When you select this option, SAS displays a help file as shown in Figure 1.5 (with the SAS Products option expanded). As with other software help systems, you can choose to search for keywords (using the Search tab), or you can scroll through the tree-like list to select topics of interest.

In Figure 1.5 note that the SAS Products section has been expanded to display links to help with a number of SAS program components. The options called Base SAS and SAS/STAT are references to the information discussed in this text. Specifically, details about the statistical procedures (PROCS) discussed in this book are documented in these two sections of the Help file.

FIGURE 1.5. *Help file display*



SUMMARY

This chapter provided an overview of SAS and examples of how to run an existing SAS program. In the following chapters we will discuss the components of a

SAS program, including how to enter data, how to request analyses, and how to format and read output.

EXERCISES

1.1. Enter code and run a SAS program.

Enter the following SAS code into the Editor window and run the program. Make sure the Editor window is clear of all other information before you begin.

```

DATA TEMP;
INPUT ID SBP DBP SEX $ AGE WT;
DATALINES;
1 120 80 M 15 115
2 130 70 F 25 180
3 140 100 M 89 170
4 120 80 F 30 150
5 125 80 F 20 110
;
RUN;
PROC PRINT DATA=TEMP;
TITLE 'Exercise 1.1 - Your Name';
RUN;
PROC MEANS;
RUN;

```

Include your name in the TITLE line. Observe the results in the Output window. To save the output to a file, make sure the Output window is active and select File ⇒ Save As, or you can select and copy the contents of the Output window and paste the output into an e-mail or word processor document. (We recommend that you change the output font to Courier or Courier New if you paste the output into a full-fledged word processor such as Microsoft Word.)

Hints for running this program:

- a. Begin SAS.
- b. Enter the SAS code into the Editor window. Pay attention to the color coding to make sure all of your code has been entered correctly.
- c. Before running the program, save the SAS code under the name C:\SASDATA\MYEXERCISE1.1.SAS.
- d. Run the program. (Click the “running man” icon or select Run ⇒ Submit on the menu bar.)
- e. Check the Log window for errors.
- f. If there are errors in the Log file, proceed to the next step; if not, go to step k.
- g. Press F12 to clear the Log and Output windows if you have previously defined the F12 key. If not, in the Log and Output windows right-click and from the pop-up menu select Edit ⇒ Clear All to erase the current contents of those windows. (This is so that you won’t accidentally try to fix an old error.)

18 Getting Started

- h. Return to the Editor window.
- i. Make corrections in the Editor window and re-save the code.
- j. Go back to step d.
- k. Examine the output.
- l. You may optionally print, save, or e-mail the output as directed by your instructor.
- m. Optionally end SAS.

1.2. Change the contents of a SAS program.

Enter the following SAS code into the Editor window and run the program. Make sure the Editor, Log, and Output windows are clear of all other information before you begin. (You can clear the Log and Output windows by pressing F12 if you have previously defined that key.)

```
DATA RECOVERY;  
INPUT LNAME $ RECTIME;  
DATALINES;  
JONES 3.1  
SMITH 3.6  
HARRIS 4.2  
MCCULLEY 2.1  
BROWN 2.8  
CURTIS 3.8  
JOHNSTON 1.8  
;  
RUN;  
PROC PRINT DATA=RECOVERY;  
Title 'Exercise 1.2 - Your Name';  
RUN;  
PROC MEANS DATA=RECOVERY;  
RUN;
```

Include your name in the TITLE line. Observe the results in the Output window.