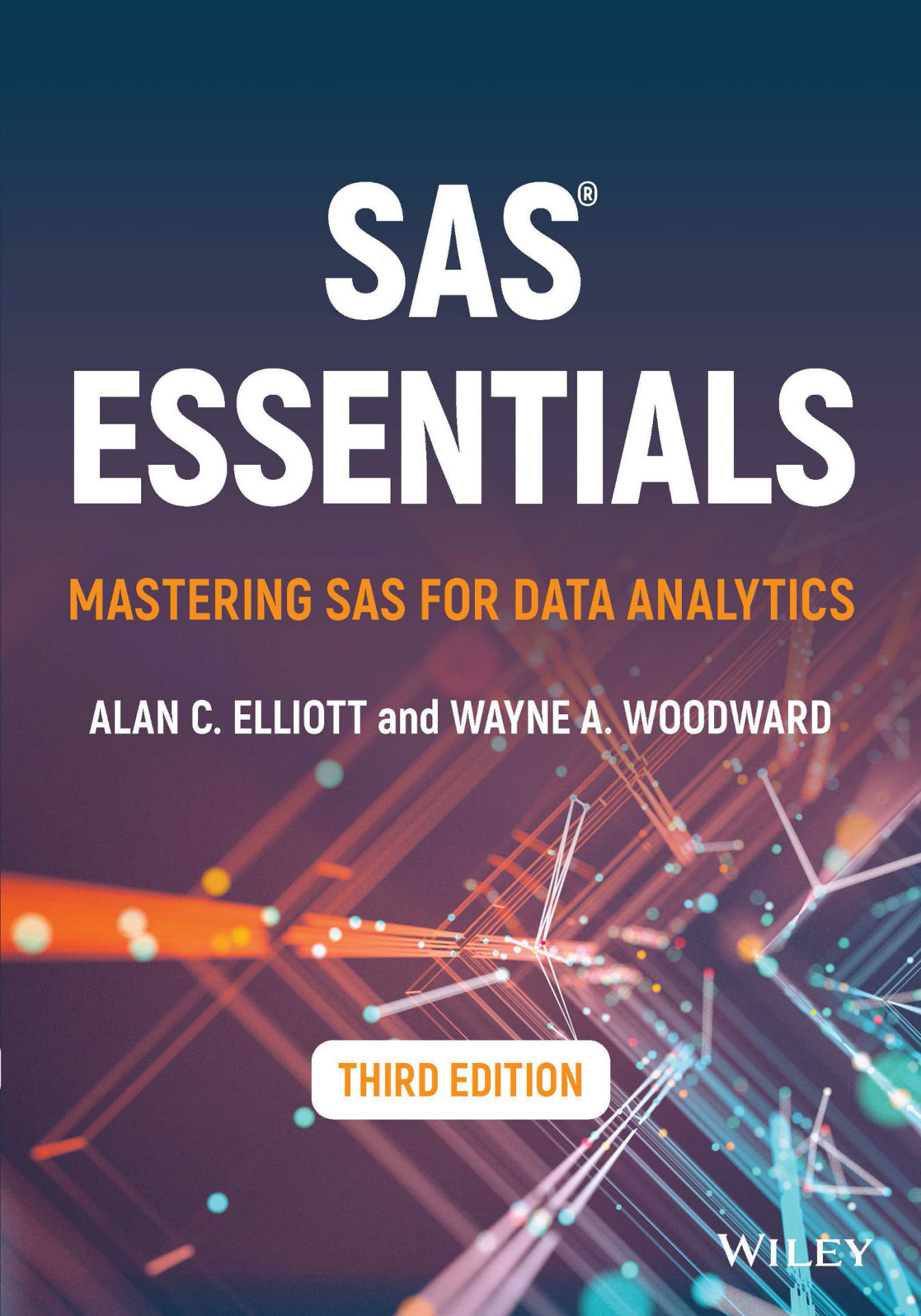

SAS®
ESSENTIALS
MASTERING SAS FOR DATA ANALYTICS
ALAN C. ELLIOTT and WAYNE A. WOODWARD
THIRD EDITION
WILEY

SAS® ESSENTIALS

SAS® ESSENTIALS

Mastering SAS for Data Analytics

Third Edition

ALAN C. ELLIOTT and WAYNE A. WOODWARD

WILEY

This edition first published 2023

Edition History
John Wiley & Sons (2e, 2015)

Registered Office
John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA

Editorial Office
111 River Street, Hoboken, NJ 07030, USA

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Limit of Liability/Disclaimer of Warranty
In view of ongoing research, equipment modifications, changes in governmental regulations, and the constant flow of information relating to the use of experimental reagents, equipment, and devices, the reader is urged to review and evaluate the information provided in the package insert or instructions for each chemical, piece of equipment, reagent, or device for, among other things, any changes in the instructions or indication of usage and for added warnings and precautions. While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Library of Congress Cataloging-in-Publication Data Applied for:
Paperback ISBN: 9781119901617

Cover Design: Wiley
Cover Image: © piranka/Getty Images

Typeset in 10/12pt Times by Straive, Pondicherry, India.

CONTENTS

ABOUT THE AUTHORS' WEBSITE

This book is accompanied by the authors' website:

https://www.alanelliott.com/SASED3/

This site will contain links to:

- The data and program files referenced in the book
- PowerPoint files for instructors
- Teaching videos that relate to the book's content
- Quick tip pages referenced in the book
- Future help or exercises related to the book's content

PART I

DATA MANIPULATION AND THE SAS® PROGRAMMING LANGUAGE

1

GETTING STARTED

LEARNING OBJECTIVES

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

Data analytics is the process of gathering and using data to understand some process, predict a future outcome, or make a decision. The SAS system is a powerful software program designed to give data analysts 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: (1) how to get data into SAS and ready for analysis using the SAS programming language and (2) how to perform the desired data analysis.

- Part 1 (Chapters 2–9) shows how to get data into SAS and prepare it for analysis.
- Part 2 (Chapters 10–20) shows how to use data to perform statistical analyses, create graphs, and produce reports.

SAS® Essentials: Mastering SAS for Data Analytics, Third Edition. Alan C. Elliott and Wayne A. Woodward.

Companion website: https://www.alanelliott.com/SASED3/

This chapter introduces you to the SAS system's use in the Microsoft Windows environment and provides numerous Hands-On examples of how to use SAS to analyze data.

1.1 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, Linux, or Windows. This book is based on using SAS in a Windows environment where you have the software installed on your local computer. The vast majority of the content in this book will apply to any SAS computer environment. However, we will include occasional references to differences that are present in other operating systems. Most of the differences between versions have to do with file references. Moreover, there are multiple ways to use SAS, notably the programming or Enterprise Guide approaches. This book teaches the programming approach that offers the user the most flexibility. Before discussing the SAS program, we'll review some of the basic file-handling features of Windows. There is a free version of SAS called OnDemand for Academics: SAS Studio for Learning Purposes. Appendix E discusses how to install this version and how it can be used to run many of the examples in this book.

1.1.1 Creating a Folder for Storing Your SAS Files

As 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 in this book in a folder named `SASDATA`. In fact, we recommend that you do this NOW!. The examples in this text will assume that all example files used in this book are in a folder whose file path is `C:\SASDATA`. You can adapt the examples by using an appropriate file path for other operating systems. Table 1.1 shows several ways to reference a file.

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

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

1. Download the example files from the following web site: `http://www.alanelliott.com/sas`
2. Follow the installation instructions provided on the web site for the 3rd edition of this book. Doing so creates the `C:\SASDATA` folder on your computer and copies the example files to that folder. (The web site also includes updates concerning the information in this book and other helpful resources.)

TABLE 1.1 Accessing Files in Various Operating Systems

Operating System	Example File Reference
Windows	`C:\SASATA\myfile`
Windows/Cytrix	`\\CLIENT\C$\SASDATA\myfile`
UNIX	`/home/sasdata/myfile`
Open VMS	`[username.sasdata]myfile`
z/OS	`SASDATA.myfile`

You can also use SAS on an Apple Mac. To do so, we recommend that you install a virtual Windows environment using commercially available programs such as Bootcamp or Parallels and install SAS in the Windows environment. You can also run SAS OnDemand for Academics on Windows, Mac, or Linux environment. (Refer to Appendix E in this book or search the web for "SAS OnDemand for Academics" for more information.

> The examples in the book are designed to use data and sample programs 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 or on a network drive – just remember to adjust the file names and file paths given in the examples in this book accordingly if you store your files in a location other than `C:\SASDATA`.

1.1.2 Beginning the SAS Program

Since there are multiple ways to use SAS, launching SAS may differ according to your installation and operating system. For a typical Windows environment, where there is a SAS icon (or tile) on your 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.

(We don't recommend that you click on a .SAS file to launch SAS because it may not open SAS in the way you want. Doing so may open SAS in the Enhanced Editor, Enterprise Guide, or Universal Viewer. (It depends on how SAS was installed on your computer.) This book uses the SAS Windows Enhanced Editor as the primary interface for using SAS. Refer to SAS documentation for information about Enterprise Guide.

1.1.3 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.)

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 visible on the normal opening screen. These include the **Graph** and **Results Viewer** windows. To open a SAS window that is not currently visible, click its tab at the bottom of the screen. (The Output tab relates to a window that is not often used and we will not discuss it here.) The following is a brief description of the windows that are used in this text.

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 Window** highlights errors in red. You should also look for warnings and

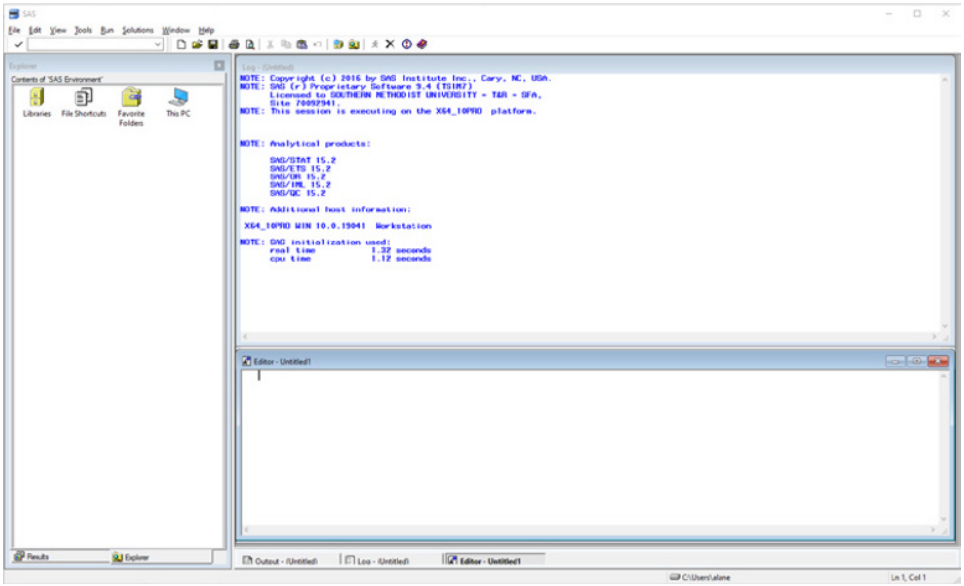


Figure 1.1 Initial SAS screen. (Source: Created with SAS® software, Version 9.4. Copyright 2022, SAS Institute Inc., Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute Inc., Cary, NC)

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 Window** (which you can display by clicking the **Explorer tab**) displays the available SAS libraries where SAS data files are stored. A SAS library is a nickname for an actual physical location on disk, such as `C:\SASDATA`. This will be described in detail in Chapter 3.

Graph: If your SAS program creates graphics output, SAS will display a **Graph Window tab**. Click that tab to view graphics results.

Results Viewer: Beginning with SAS 9.3, the results of analysis appear in this viewer. It will appear the first time you run an analysis that creates output.

> Do not close any of the windows that make up the SAS interface. Move from window to window by clicking on the tabs. 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. (Or restart the program.)

1.2 YOUR FIRST SAS ANALYSIS

Now that you've 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 Section 1.1.1 and do so.) Once you have downloaded the example files to your computer, continue with this chapter.

The following steps show you how to open a SAS program file, perform a simple analysis using 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 1.1

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

1. Launch SAS. 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.
3. Examine the opened file, the contents of which appear in the **Editor Window**. Maximize the **Editor Window** so that you can see more of the program code if necessary. Figure 1.2 shows the **Editor Window** maximized. We'll learn more in the latter chapters about what these lines of code mean.

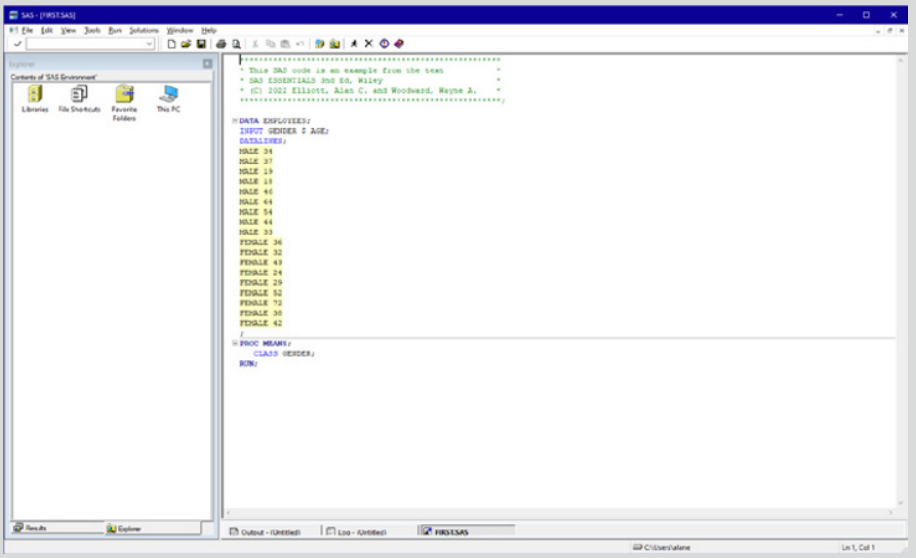


Figure 1.2 The FIRST.SAS file opened

4. Run the SAS job. There are three ways to run (sometimes called submit) your SAS code. (1) On the menu bar, select **Run → Submit**, (2) click the "running man" icon on the toolbar, or (3) press F8. Perform any of these to run this code. The SAS

(*Continued*)

(*Continued*)

instructions in the file are carried out (or, in computer terminology, the commands are *executed*). The SAS program creates an analysis output that automatically appears in the **Results Viewer**.

5. View the results shown in Figure 1.3.
   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. Note the **Results Window** (as opposed to the **Results Viewer**) on the left of the screen illustrated 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 in this **Results Window**.

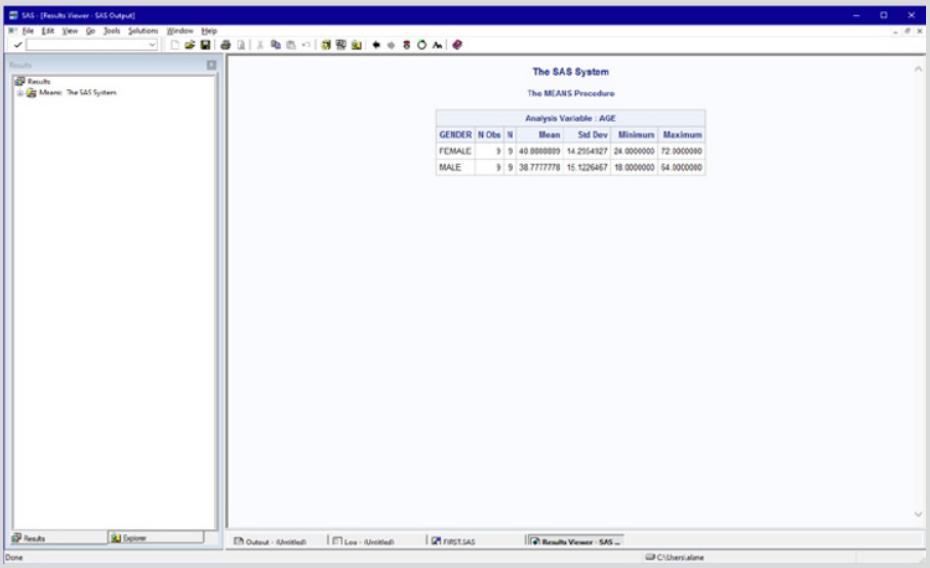


Figure 1.3 Results for FIRST.SAS

6. Print the output. Make sure the **Results Viewer** is active by clicking anywhere in the **Results Viewer**. Print the contents of the viewer 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 **Results Viewer**, select **File → Save As** on the menu bar and enter an appropriate name for the output file, such as `FIRST OUTPUT`. The file will be saved as `FIRST OUTPUT.MHT`. (The .MHT extension refers to a single file web page file and is automatically added to the file name. If you subsequently open this file, it will open into your default web browser.)
8. Save the SAS program code (instruction file) that created the output by first going to the **Editor Window** that contains the `FIRST.SAS` program code. With the program on the screen, select **File → Save As** and enter a name such as `MYFIRST`. Make sure

you are saving the code in the C:\SASDATA folder. This saves the file as MYFIRST.SAS. (The .SAS extension is automatically added to the file name.)

9. Examine the SAS **Log Window**. Click on the **Log Window** tab at the bottom of your screen and note that it contains a report on the SAS program code you just ran. It will include statements such as the following:

```
NOTE: Writing HTML Body file: sashtml.htm
NOTE: There were 18 observations read from the data set
WORK.EMPLOYEES.
```

This indicates that the output is in an html format (web file format) and that the data set contained 18 records.

10. Clear the contents of the **Log Window** by right-clicking and selecting **Edit → Clear All** from the pop-up menu. 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 the window that is currently active. Thus, if the **Results Viewer** is active, it saves the output as an *.MHT file; if the **Log Window** is active, it saves the contents of that window as an *.LOG file; and if the **Editor Window** is the active window, it saves the code as an *.SAS (SAS program code) file.

That's it! You've run your first SAS program. 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 **Log** and **Editor Windows** so that you will not mix up information 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 Section 1.4.2, we will show a way to shorten this process.

HANDS-ON EXAMPLE 1.2

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

1. In the SAS **Editor Window**, open the program file SECOND.SAS. (It should be in your C:\SASDATA folder.) The SAS code looks like this:

```
* PUT YOUR TITLE ON THE NEXT LINE;
DATA EXAMPLE;
INPUT AGE @@;
DATALINES;
12 11 12 12 9 11 8 8 7 11 12 14 9 10 7 13
6 11 12 4 11 9 13 6 9 7 13 9 13 12 10 13
11 8 11 15 12 14 10 10 13 13 10 8 12 7 13
11 9 12
```

(*Continued*)

(*Continued*)

```
;
PROC MEANS DATA=EXAMPLE;
     VAR AGE;
RUN;
```

Run the SAS job using any of the methods described in the previous example and observe the output. Note that the title at the top of the output page is "The SAS System."

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 note how the title has changed in the output.

1.3 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 as follows:

1. Define a data set in SAS using the `DATA` step (which begins with the keyword `DATA`.) In this case, the data values are a part of the code (although it is not always the case.) The data values to be used in this analysis follow the keyword `DATALINES`.
2. Once you have a data set of values, you can tell SAS what analysis to perform using a procedure (`PROC`) statement. In this case, the keywords `PROC MEANS` initiate the "MEANS" procedure.
3. Run the program and observe the output (in the **Results Viewer**).

Figure 1.4 illustrates this process.

This book teaches you how to create and use SAS by illustrating SAS techniques of data handling and analysis through a number of sample SAS programs.

To exit 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.)

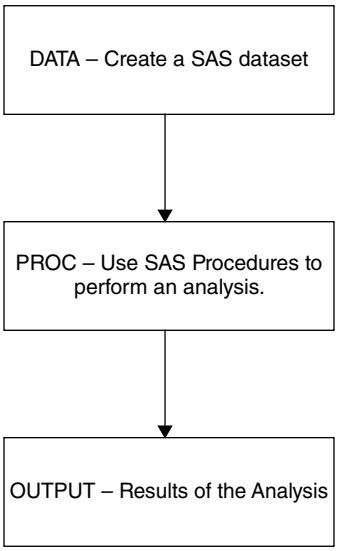


Figure 1.4 How SAS works

1.4 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`, `DATALINES`, `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. Moreover, 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 specify 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. (For some procedures, you should also end with the `QUIT;` statement. Those will be discussed when they are mentioned in the text.)
- **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 Window 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 latter 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.** Note how the example programs in this book are structured – how some statements are indented for easy reading and how the code contains comments. (Lines that begin with an asterisk (*) are comments that are ignored when the program is run. Comments are discussed more in Chapter 5.)

Now that you have experience running simple programs in SAS, you will want to know more about the syntax and conventions for entering and running a SAS program. The following is a brief introduction to some of the SAS program requirements you need to know.

1.4.1 Using the SAS Enhanced Editor

You may have noted that different parts of the coding in `FIRST.SAS` or `SECOND.SAS` appear in various colors and in bold font. 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 the SAS Enhanced Editor:

Green is used for comments.
Bold 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 background is used to highlight 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 before you attempt to run the program.

TABLE 1.2 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 this Keys Window
F10	Not defined	
F11	`COMMAND BAR`	Moves cursor to command bar
F12	Not defined	

A boundary line separates steps (parts) of a SAS program. In the SECOND.SAS example, note that a minus sign in a small box appears on the screen next to the keyword DATA (and another one next to PROC MEANS.) Clicking on this box compresses the step to its first line, and is a way to temporarily hide sections of code. When you run the code, the program still "sees and acts upon" what is in the compressed section.

1.4.2 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 1.2 lists some of the SAS function keys that you can use throughout this book.

You can also customize these keys. To display the full set of function key commands, on the menu bar select **Tools → Options → Keys**.

HANDS-ON EXAMPLE 1.3

In earlier examples, you cleared the information from the **Log Window** manually. In this example, you will create a new F12 function key command that will clear the **Log Window** and the **Results Viewer** and return you to the **Editor Window**. The F12 key you're about to define uses the SAS commands:

- **`CLEAR LOG`** – clears the contents of the **Log Window**.
- **`ODSRESULTS;CLEAR;`** – clears the contents of the **Results Viewer** (and closes that window). Note that there is no blank between `ODS` and `RESULTS`.
- **`WPGM;`** returns you to the WPGM (i.e., to the active window most recently edited).

Following these instructions shows how to create a new F12 function key definition.

1. Press the F9 function key to display the **Keys Window**. Table 1.2 shows some of the key definitions that appear on your screen.

(*Continued*)

(*Continued*)

2. Next to the blank F12 option, enter

```
CLEAR LOG; ODSRESULTS; CLEAR; WPGM;
```

   Press the **Enter** key to lock in the new command.
3. Exit the **`Keys` Window** (by clicking on the small x at the top right of the window) and try out this function key by re-running one of the previous example programs. With the output displayed on the screen, press F12. The **Results Viewer** will be cleared and closed, the **Log Window** 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 Window** and **Results Viewer** without deleting the program code. This new command will be used in some of our future examples. (If you press F12 and your **Editor Window** does not reappear, press it again, or press F5 to display the **Editor Window**.)

Use the F12 SAS function key you created to clear the contents of the **Results Viewer** and **Log Window** between examples.

1.4.3 Using the SAS Menus

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

The SAS window you will use most is the **Enhanced 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, importing and exporting 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.
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 Window or Output Viewer**.
Help: Contains options for the SAS help system as well as online documentation. (See the Section 1.4.5.)

1.4.4 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 or *.docx extension indicates a Microsoft Word file).

SAS Code File (*filename.sas*): This is an ACSII 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 Results File (*filename.mht* or *filename.html*): This file contains the web-formatted output such as that displayed in the **Results Viewer**. HTML stands for Hyper-Text Markup Language and is the common language of Internet web files. MHT is short for MHTML and stands for Microsoft (or MIME) Hypertext Archive file. It is a type of HTML file that contains the entire html-coded information in a single file (whereas HTML files may access external files for some components such as graphs.)

SAS Data File (*filename.sas7bdat*): This file contains a SAS data set that includes 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* or *filename.xlsx*): The data in a Microsoft Excel file (when properly formatted into a table of columnar data) can be imported into SAS. (We'll discuss data file types that can be imported into SAS in Chapter 3.)

1.4.5 Getting SAS Help

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.

SAS help is also available on the web in a number of forms. SAS Institute provides many of their reference manuals on the web (as pdf files), there are a number of papers presented at SAS conferences on the web, and a number of independent web sites that include SAS examples and help. A search for a particular SAS topic (i.e., "SAS ANOVA") will typically bring up a large number of links.

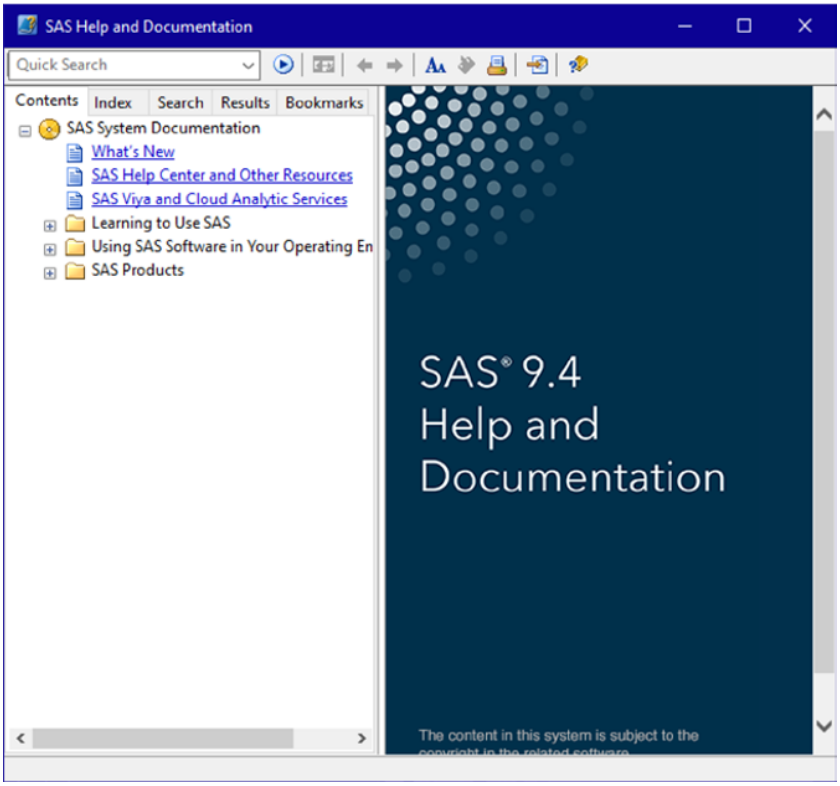


Figure 1.5 Help file display

1.5 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 the output.

EXERCISES

1.1 Enter code and run a SAS program.
Make sure you're in the **Editor Window**. **Choose File → New 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 **Results Viewer**. To save the output to a file, make sure the **Results Viewer** is active and select **File → Save As**, or you can select and copy the contents of the **Results Viewer** and paste the output into an e-mail or word processor document. Hints for running this program are as follows:

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**, observe where the error occurred. SAS will usually point out the code that produced the error. If there were errors, proceed to the next step; if not, skip to step k.
g. After determining what changes should be made to your code, press F12 to clear the **Log Window** and **Results Viewer** if you have previously defined the F12 key. Or go to the **Log Window** and right-click, and choose **Edit → Clear All.** (This clears the error messages out of the **Log Window** so that you won't get confused by seeing an old error.)
h. Return to the **Editor Window**.
i. Make corrections in the **Editor Window** to fix the error you discovered in step f. Re-save the code.
j. Go back to step d.
k. Examine the output in the **Results Viewer**.
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.**
Make sure you're in the **Editor Window**. Choose **File → New Program**. Enter the following SAS code into the **Editor Window** and run the program. Make sure the **Editor Window**, **Log Window**, and **Results Viewer** are clear of all other information

before you begin. (You can clear the **Log Window** and **Results Viewer** by pressing F12 if you have previously defined that key.)

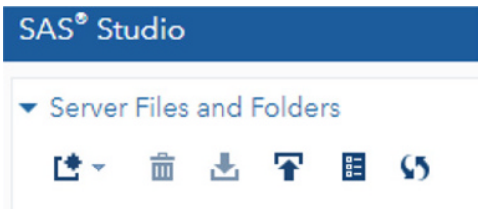
```
DATA RECOVERY;
INPUT LNAME $ RECTIME;
DATALINES;
JONES 3.1
SMITH 3.6
HARRIS 4.2
MCCULLEY 2.1
BROWN 2.8
CURTIS 3.8
JOHNSTON 1.8
;
RUN;
PROC PRINT DATA=RECOVERY;
Title 'Exercise 1.2 - Your Name';
RUN;
PROC MEANS DATA=RECOVERY;
RUN;
```

Include your name in the `TITLE` line. Observe the results in the **Results Viewer**. (You may optionally print, save, or e-mail the output as directed by your instructor. These instructions will be assumed from this point on and will not be included in future exercises.)

1.3 (Optional) If you are using SAS Studio, here is how you could access the `FIRST.SAS` example. We'll not provide this detail for all exercises, but this is given here for those who are using SAS Studio.

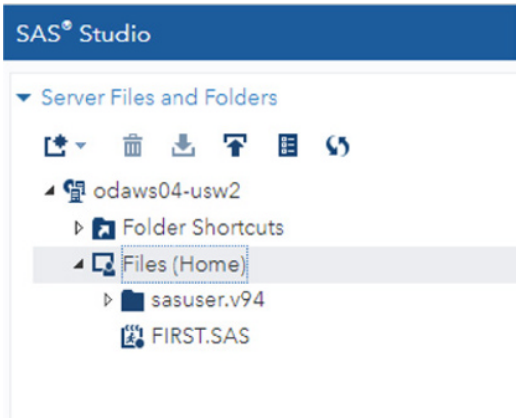
a. Begin SAS Studio (See Appendix E for information on installing and beginning SAS Studio.)

b. Locate the upload icon (⬆) at the top left of the SAS Studio screen.

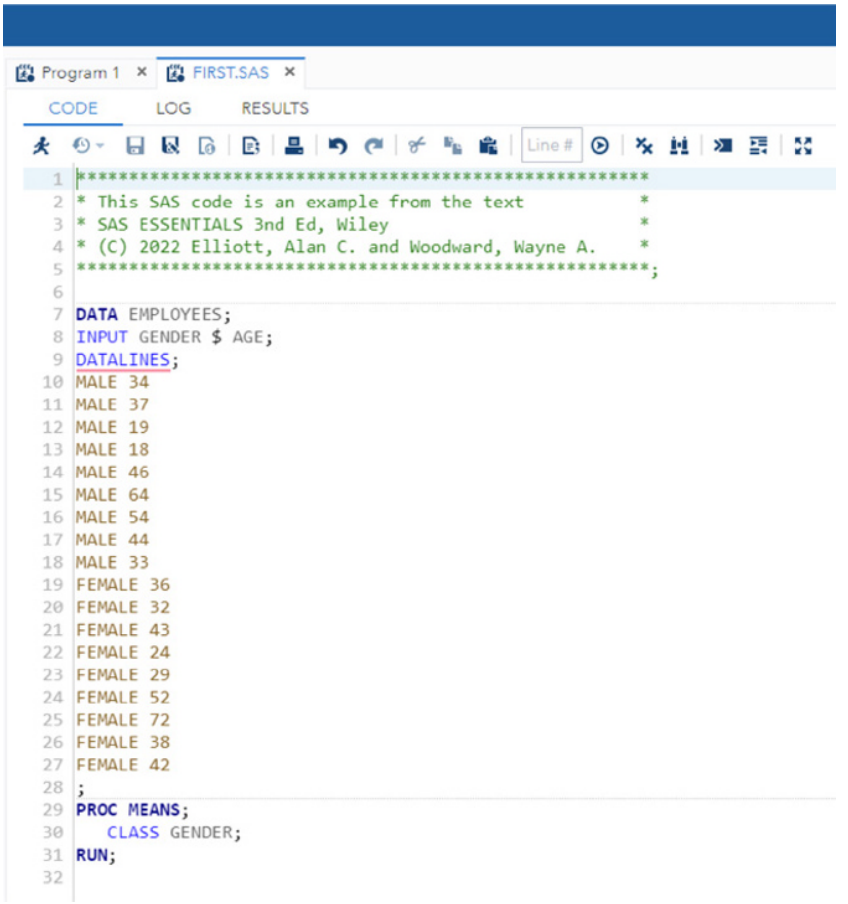


c. In the **Upload Files** dialog box, click **Choose Files**. From the **Windows File Selector,** go to `C:\SASDATA` and select `FIRST.SAS`. Click Open.

d. You will return to the **Upload Files** dialog box. This time, choose **Upload**. This puts the `FIRST.SAS` files into the designated SAS Studio library (usually \home\ number). The `FIRST.SAS` file should now appear in the Files (Home) list (as shown here.)



e. Double click on `FIRST.SAS` and the code will appear in the **SAS Studio Code Window**.



f. Click the running man icon to run the `FIRST.SAS` program

g. Compare the results to what you saw in Figure 1.3

2

GETTING DATA INTO SAS®

LEARNING OBJECTIVES*

- To enter data using freeform list input
- To enter data using the compact method
- To enter data using column input
- To enter data using formatted input
- To enter data using the INFILE technique
- To enter multiple-line data

Before you can perform any analysis, you must have data. SAS contains powerful tools you can use to enter, import, and manage your data. This chapter introduces the most commonly used methods for getting data into SAS. Additional techniques are covered in Chapter 3.

The main way to enter and manipulate data in SAS is by using the SAS `DATA` step. This `DATA` step creates a specially formatted type of file referred to as a "SAS dataset" or "SAS data file." A SAS dataset contains the raw data plus other information. It contains variable names, how data values are read and interpreted, how values are displayed in the output, labels associated with the variable names, and other information.

SAS® Essentials: Mastering SAS for Data Analytics, Third Edition. Alan C. Elliott and Wayne A. Woodward.

Companion website: https://www.alanelliott.com/SASED3/