

1

“An honest tale speeds best
being plainly told.”

WILLIAM SHAKESPEARE, *KING RICHARD III*



CHAPTER 1

Who, What, and How

- 1.1 What This Book Covers and Who Needs It 2
- 1.2 How This Book Works 4
- 1.3 Layout of Sample Programs 6

1.1 What This Book Covers and Who Needs It

This book is for all new SAS users in business, government, academia, or anyone who will be doing data analysis using the SAS System. Readers need no prior experience with SAS software, but people who have some experience may still find this book useful for learning techniques they missed or as an easy-to-use reference.

This book introduces readers to the SAS language with lots of practical examples, clear and concise explanations, and as little technical jargon as possible. Most of the features covered here come from Base SAS, which contains the core of features used by all SAS programmers. The only exception is a brief chapter introducing SAS/STAT software. Once you learn to use Base SAS, you will be ready to learn other SAS products (such as SAS/AF, SAS/ETS, SAS/FSP, SAS/OR, or SAS/QC software) if you wish.

We included every feature of Base SAS that a beginning user is likely to need. Some people will be surprised because certain topics, such as arrays, are normally considered advanced. But they appear here because sometimes new users need them. However, that doesn't mean that you need to know everything in this book. On the contrary, this book is designed so you can read just those sections you need to solve your problems. Even if you read this book from cover to cover, you may find yourself returning to refresh your memory as new programming challenges arise.

This book starts with general concepts of the SAS language and then addresses common tasks such as

- ▶ running SAS programs
- ▶ reading raw data
- ▶ modifying data
- ▶ sorting, summarizing, and printing data
- ▶ combining data sets
- ▶ applying basic statistical procedures
- ▶ debugging SAS programs.

In the chapter on statistics, we cover four commonly used procedures for univariate statistics, correlation, regression, and analysis of variance. After you master the procedures in this book, the statements, options, and output available in other statistical procedures will look familiar.

Unfortunately, a book of this type cannot provide a thorough introduction to statistical concepts such as degrees of freedom, or crossed and nested effects. We assume that readers who are interested in statistical computing already know something about statistics. People who want to use statistical procedures but are unfamiliar with these concepts should seek out an introductory statistics text or, better yet, take a course in statistics.

The chapter on debugging is particularly unique and important. New users are sometimes intimidated by error messages, but there is no need to be. This chapter includes all the bugs that people commonly encounter, their possible causes, and how to fix them. Here you will also find advice on programming techniques that will help you to avoid getting bugs in the first place.

The appendices include more in-depth information for people with specific interests or experience:

- ▶ where to go for more information
- ▶ overview of SAS products
- ▶ coming to SAS from SPSS
- ▶ coming to SAS from a programming language
- ▶ coming to SAS from SQL.

To use this book you need no prior knowledge of SAS, but you must know something about your computer and local operating system. The SAS System behaves virtually the same from one operating system to another, but some differences are unavoidable. For example, every operating system has a different way of storing and accessing files. Some operating systems have more of a capacity for interactive computing than others. And the exact commands for printing files may be different even for two physically identical computers because of different local customs. This book addresses operating systems as much as possible, but no book can answer every question about your local operating system. You must have either a working knowledge of your operating system or someone you can turn to with questions.

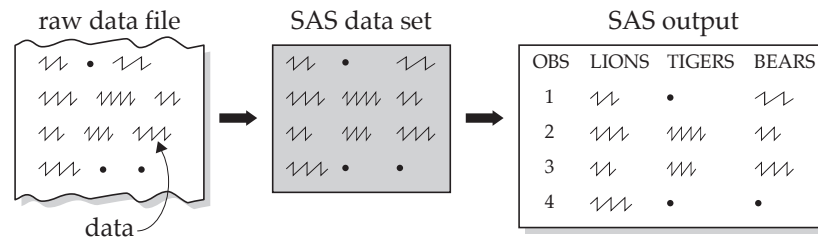
This little book is not a replacement for the many SAS reference manuals. However, after reading this book, you will be able to go to the reference manuals and confidently find any in-depth information you need.

Last, we tried to make this book as readable as possible and, we hope, even enjoyable. Once you master the contents of this small book you will no longer be a beginning SAS programmer.

1.2 How This Book Works

Modular sections Our goal in writing this book is to make learning SAS as easy and enjoyable as possible. Let's face it, SAS is a big topic. You may have already spent some time scratching your head in front of a shelf full of SAS manuals. We can't condense that entire shelf into this little book, but we can condense topics into short, readable sections. This entire book is composed of two-page sections, each section a complete topic. This way, you can easily skip over topics which do not apply to you. Of course, we think every section is important, or we would not have included it. You probably don't need to know everything in this book, however, to complete your job. By presenting topics in short digestible sections, we believe that learning SAS will be easier and more fun—like eating three meals a day instead of one giant meal a week.

Graphics Wherever possible, graphics either identify the contents of the section or help explain the topic. A box with rough edges indicates a raw data file, and a box with nice smooth edges indicates a SAS data set. The squiggles inside the box indicate data—any old data—and a period indicates a missing value. The arrow between boxes of these types means that the section explains how to get from data that look like one box to data that look like the other. Some sections have graphics which depict printed output. These graphics look like a stack of papers with variable names printed on the top of the page and observation numbers shown on the left.




ates a missing value. The arrow between boxes of these types means that the section explains how to get from data that look like one box to data that look like the other. Some sections have graphics which depict printed output. These graphics look like a stack of papers with variable names printed on the top of the page and observation numbers shown on the left.


Syntax Some sections show the general form of particular statements (commonly referred to as syntax), and may also give example statements. When showing syntax, what is written in italics is a description of what goes in that spot in the statement, not what you actually type. If it is not in italics, then it is an example of what you could put there. SAS keywords are written in uppercase letters, and must be spelled exactly as shown.


The following shows two VAR statements. The first shows the syntax, or general form of the statement, while the second shows an example of an actual statement as it might appear in a SAS program. Notice that the keyword VAR is the same in both statements and that the descriptive term *variable-list* in the syntax statement has been replaced with actual variable names in the example statement.


Syntax	Example
VAR <i>variable-list</i> ;	VAR lions tigers;

Cross-references Occasionally, you may need more detail than would be appropriate for this book. To help you at those times, each section ends with cross-references. These cross-references are pointers to the SAS reference manuals where you can find more information on the section's topic. Each cross-reference includes a symbol (call it an icon if you like) for the SAS reference manual and the index item for that manual. By listing the index item rather than the page number or section name, we hope to encourage you to use the index. The index is a valuable tool when you are using SAS as you will always have to look things up. Unless you get very good at flipping to the right page, you will need to use the index. The following symbols are used for the SAS reference manuals:

*SAS Language:Reference,
Version 6, First Edition* 

*SAS/STAT User's Guide, Version 6,
Fourth Edition, Volumes 1 and 2* 

*SAS Procedures Guide,
Version 6, Third Edition* 

*SAS Companion for the
(your) Environment¹* 

A typical cross-reference would look like this:

INPUT statement 
list input

This means to look in the index of *SAS Language: Reference, Version 6, First Edition* for *INPUT statement* and then for *list input*.

¹There is a SAS Companion reference manual for each operating system or environment. SAS tries to make things work the same in every environment, but sometimes this is impossible. The SAS Companion manuals cover topics specific to your environment.

1.3 Layout of Sample Programs

This book contains many sample SAS programs, each complete and executable. Programs are formatted in a way which makes them easy for you to read and understand. You do not have to format your programs this way, as SAS is very flexible, but attention to some of these details will make your programs easier to read. Easy-to-read programs are time savers for you, or the consultant you hire at \$100 per hour, when you need to go back and decipher the program months or years later.

Each sample program is formatted in the following way:

Capitalization All SAS keywords appear in uppercase letters. A keyword is an instruction to SAS and must be spelled correctly. Anything in lowercase letters is something that the programmer has made up: a variable name, a name for a SAS data set, a comment, or a title. For example, the following SAS statement shows the keyword INPUT in upper case and the variable names lions and tigers in lower case:

```
INPUT lions tigers;
```

SAS doesn't care if statements are in upper or lower case. This is an aid for you so you can easily see which words are SAS keywords and which are not.

Indentation The different parts of the program are shown by indenting all statements after the first in a step. This is a simple way to make your programs more readable, and it's a good habit to form. SAS doesn't really care where statements start or even if they are all on one line. In the following program, the INFILE and INPUT statements are indented, indicating that they belong with the DATA statement:

```
* Read animals' weights from file. Print the results.;
DATA animals;
    INFILE 'zoo.dat';
    INPUT lions tigers;
PROC PRINT;
RUN;
```

Comments When appropriate, programs in this book have comments to help explain what the program does. The comments used in this book start with an asterisk (*) and end with a semicolon (;). It doesn't matter what's between the asterisk and the semicolon, SAS doesn't look at it. You could put your favorite cookie recipe in there if you want. However, comments are usually used to annotate the program, making it easier for someone to read your program and understand what you have done and why. For example, the preceding program starts with a comment statement describing the purpose of the program.

Another style of comment starts with a slash asterisk (/*) and ends with an asterisk slash (*). This style of comment, unlike the asterisk-semicolon style, can be used within a SAS statement and can contain embedded semicolons. The following program shows this style of comment statement:

```
/* Read animals' weights from file */  
DATA animals;  
  INFILE 'zoo.dat';  
  INPUT lions tigers;  
PROC PRINT; /* Print the results */  
RUN;
```

Since some operating systems interpret a slash asterisk (/*) in the first column as the end of a job, be careful when using this style of comment not to place it in the first column. For this reason, we chose the asterisk-semicolon style of comment for this book.

