

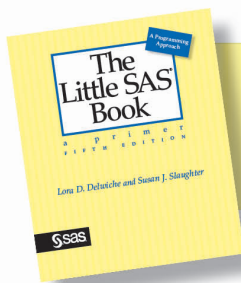
A Programming
Approach

The Little SAS[®] Book

a p r i m e r
F I F T H E D I T I O N

Lora D. Delwiche and Susan J. Slaughter





From *The Little SAS® Book, Fifth Edition*.
Full book available for purchase [here](#).

Contents

Acknowledgments	ix
Introducing SAS Software	x
About This Book	xi
What's New	xiv

Chapter 1	Getting Started Using SAS Software	1
1.1	The SAS Language	2
1.2	SAS Data Sets	4
1.3	DATA and PROC Steps	6
1.4	The DATA Step's Built-in Loop	8
1.5	Choosing a Mode for Submitting SAS Programs	10
1.6	Windows and Commands in the SAS Windowing Environment	12
1.7	Submitting a Program in the SAS Windowing Environment	14
1.8	Reading the SAS Log	16
1.9	Viewing Your Results	18
1.10	SAS Data Libraries	20
1.11	Viewing Data Sets in the Viewtable Window	22
1.12	Viewing the Properties of Data Sets with SAS Explorer	24
1.13	Using SAS System Options	26
Chapter 2	Getting Your Data into SAS	29
2.1	Methods for Getting Your Data into SAS	30
2.2	Entering Data with the Viewtable Window	32
2.3	Reading Files with the Import Wizard	34
2.4	Telling SAS Where to Find Your Raw Data	36
2.5	Reading Raw Data Separated by Spaces	38
2.6	Reading Raw Data Arranged in Columns	40
2.7	Reading Raw Data Not in Standard Format	42
2.8	Selected Informats	44

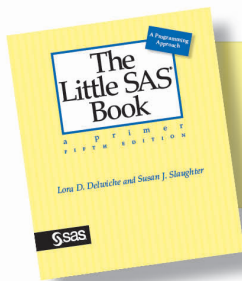
2.9	Mixing Input Styles	46
2.10	Reading Messy Raw Data	48
2.11	Reading Multiple Lines of Raw Data per Observation	50
2.12	Reading Multiple Observations per Line of Raw Data	52
2.13	Reading Part of a Raw Data File	54
2.14	Controlling Input with Options in the INFILE Statement	56
2.15	Reading Delimited Files with the DATA Step	58
2.16	Reading Delimited Files with the IMPORT Procedure	60
2.17	Reading Excel Files with the IMPORT Procedure	62
2.18	Temporary versus Permanent SAS Data Sets	64
2.19	Using Permanent SAS Data Sets with LIBNAME Statements	66
2.20	Using Permanent SAS Data Sets by Direct Referencing	68
2.21	Listing the Contents of a SAS Data Set	70
Chapter 3	Working with Your Data	73
3.1	Creating and Redefining Variables	74
3.2	Using SAS Functions	76
3.3	Selected SAS Character Functions	78
3.4	Selected SAS Numeric Functions	80
3.5	Using IF-THEN Statements	82
3.6	Grouping Observations with IF-THEN/ELSE Statements	84
3.7	Subsetting Your Data	86
3.8	Working with SAS Dates	88
3.9	Selected Date Informats, Functions, and Formats	90
3.10	Using the RETAIN and Sum Statements	92
3.11	Simplifying Programs with Arrays	94
3.12	Using Shortcuts for Lists of Variable Names	96
Chapter 4	Sorting, Printing, and Summarizing Your Data	99
4.1	Using SAS Procedures	100
4.2	Subsetting in Procedures with the WHERE Statement	102
4.3	Sorting Your Data with PROC SORT	104
4.4	Changing the Sort Order for Character Data	106
4.5	Printing Your Data with PROC PRINT	108

4.6	Changing the Appearance of Printed Values with Formats	110
4.7	Selected Standard Formats	112
4.8	Creating Your Own Formats Using PROC FORMAT	114
4.9	Writing Simple Custom Reports	116
4.10	Summarizing Your Data Using PROC MEANS	118
4.11	Writing Summary Statistics to a SAS Data Set	120
4.12	Counting Your Data with PROC FREQ	122
4.13	Producing Tabular Reports with PROC TABULATE	124
4.14	Adding Statistics to PROC TABULATE Output	126
4.15	Enhancing the Appearance of PROC TABULATE Output	128
4.16	Changing Headers in PROC TABULATE Output	130
4.17	Specifying Multiple Formats for Data Cells in PROC TABULATE Output	132
4.18	Producing Simple Output with PROC REPORT	134
4.19	Using DEFINE Statements in PROC REPORT	136
4.20	Creating Summary Reports with PROC REPORT	138
4.21	Adding Summary Breaks to PROC REPORT Output	140
4.22	Adding Statistics to PROC REPORT Output	142
4.23	Adding Computed Variables to PROC REPORT Output	144
4.24	Grouping Data in Procedures with User-Defined Formats	146
Chapter 5	Enhancing Your Output with ODS	149
5.1	Concepts of the Output Delivery System	150
5.2	Tracing and Selecting Procedure Output	152
5.3	Creating SAS Data Sets from Procedure Output	154
5.4	Creating Text Output	156
5.5	Creating HTML Output	158
5.6	Creating RTF Output	160
5.7	Creating PDF Output	162
5.8	Customizing Titles and Footnotes	164
5.9	Customizing PROC PRINT with the STYLE= Option	166
5.10	Customizing PROC REPORT with the STYLE= Option	168
5.11	Customizing PROC TABULATE with the STYLE= Option	170

5.12	Adding Traffic-Lighting to Your Output	172
5.13	Selected Style Attributes	174
Chapter 6	Modifying and Combining SAS Data Sets	177
6.1	Modifying a Data Set Using the SET Statement	178
6.2	Stacking Data Sets Using the SET Statement	180
6.3	Interleaving Data Sets Using the SET Statement	182
6.4	Combining Data Sets Using a One-to-One Match Merge	184
6.5	Combining Data Sets Using a One-to-Many Match Merge	186
6.6	Merging Summary Statistics with the Original Data	188
6.7	Combining a Grand Total with the Original Data	190
6.8	Updating a Master Data Set with Transactions	192
6.9	Writing Multiple Data Sets Using the OUTPUT Statement	194
6.10	Making Several Observations from One Using the OUTPUT Statement	196
6.11	Using SAS Data Set Options	198
6.12	Tracking and Selecting Observations with the IN= Option	200
6.13	Selecting Observations with the WHERE= Option	202
6.14	Changing Observations to Variables Using PROC TRANSPOSE	204
6.15	Using SAS Automatic Variables	206
Chapter 7	Writing Flexible Code with the SAS Macro Facility	209
7.1	Macro Concepts	210
7.2	Substituting Text with Macro Variables	212
7.3	Concatenating Macro Variables with Other Text	214
7.4	Creating Modular Code with Macros	216
7.5	Adding Parameters to Macros	218
7.6	Writing Macros with Conditional Logic	220
7.7	Writing Data-Driven Programs with CALL SYMPUT	222
7.8	Debugging Macro Errors	224
Chapter 8	Visualizing Your Data	227
8.1	Concepts of ODS Graphics	228
8.2	Creating Bar Charts	230
8.3	Creating Histograms and Density Curves	232

8.4	Creating Box Plots	234
8.5	Creating Scatter Plots	236
8.6	Creating Series Plots	238
8.7	Creating Fitted Curves	240
8.8	Controlling Axes and Reference Lines	242
8.9	Controlling Legends and Insets	244
8.10	Customizing Graph Attributes	246
8.11	Creating Paneled Graphs	248
8.12	Specifying Image Properties and Saving Graphics Output	250
Chapter 9	Using Basic Statistical Procedures	253
9.1	Examining the Distribution of Data with PROC UNIVARIATE	254
9.2	Creating Statistical Graphics with PROC UNIVARIATE	256
9.3	Producing Statistics with PROC MEANS	258
9.4	Testing Means with PROC TTEST	260
9.5	Creating Statistical Graphics with PROC TTEST	262
9.6	Testing Categorical Data with PROC FREQ	264
9.7	Creating Statistical Graphics with PROC FREQ	266
9.8	Examining Correlations with PROC CORR	268
9.9	Creating Statistical Graphics with PROC CORR	270
9.10	Using PROC REG for Simple Regression Analysis	272
9.11	Creating Statistical Graphics with PROC REG	274
9.12	Using PROC ANOVA for One-Way Analysis of Variance	276
9.13	Reading the Output of PROC ANOVA	278
Chapter 10	Exporting Your Data	281
10.1	Methods for Exporting Your Data	282
10.2	Writing Files Using the Export Wizard	284
10.3	Writing Delimited Files with the EXPORT Procedure	286
10.4	Writing Microsoft Excel Files with the EXPORT Procedure	288
10.5	Writing Raw Data Files with the DATA Step	290
10.6	Writing Delimited and HTML Files Using ODS	292
Chapter 11	Debugging Your SAS Programs	295
11.1	Writing SAS Programs That Work	296
11.2	Fixing Programs That Don't Work	298

11.3	Searching for the Missing Semicolon	300
11.4	Note: INPUT Statement Reached Past the End of a Line	302
11.5	Note: Lost Card	304
11.6	Note: Invalid Data	306
11.7	Note: Missing Values Were Generated	308
11.8	Note: Numeric Values Have Been Converted to Character (or Vice Versa)	310
11.9	DATA Step Produces Wrong Results but No Error Message	312
11.10	Error: Invalid Option, Error: The Option Is Not Recognized, or Error: Statement Is Not Valid	314
11.11	Note: Variable Is Uninitialized or Error: Variable Not Found	316
11.12	SAS Truncates a Character Variable	318
11.13	SAS Stops in the Middle of a Program	320
11.14	SAS Runs Out of Memory or Disk Space	322
Appendix Coming to SAS from SQL		325
Index		331



From *The Little SAS® Book, Fifth Edition*.
Full book available for purchase [here](#).

CHAPTER 1

Getting Started Using SAS Software

- 1.1 The SAS Language 2
- 1.2 SAS Data Sets 4
- 1.3 DATA and PROC Steps 6
- 1.4 The DATA Step's Built-in Loop 8
- 1.5 Choosing a Mode for Submitting SAS Programs 10
- 1.6 Windows and Commands in the SAS Windowing Environment 12
- 1.7 Submitting a Program in the SAS Windowing Environment 14
- 1.8 Reading the SAS Log 16
- 1.9 Viewing Your Results 18
- 1.10 SAS Data Libraries 20
- 1.11 Viewing Data Sets in the Viewtable Window 22
- 1.12 Viewing the Properties of Data Sets with SAS Explorer 24
- 1.13 Using SAS System Options 26

1.1 The SAS Language

Many software applications are either menu driven, or command driven (enter a command—see the result). SAS is neither. With SAS, you use statements to write a series of instructions called a SAS program. The program communicates what you want to do and is written using the SAS language. There are some menu-driven front ends to SAS, for example SAS Enterprise Guide, which make SAS appear like a point-and-click program. However, these front ends still use the SAS language to write programs for you. You will have much more flexibility using SAS if you learn to write your own programs using the SAS language. Maybe learning a new language is the last thing you want to do, but be assured that although there are parallels between SAS and languages you know (be they English or JAVA), SAS is much easier to learn.

SAS programs A SAS program is a sequence of statements executed in order. A statement gives information or instructions to SAS and must be appropriately placed in the program. An everyday analogy to a SAS program is a trip to the bank. You enter your bank, stand in line, and when you finally reach the teller's window, you say what you want to do. The statements you give can be written down in the form of a program:

```
I would like to make a withdrawal.  
My account number is 0937.  
I would like $200.  
Give me five 20s and two 50s.
```

Note that you first say what you want to do, then give all the information the teller needs to carry out your request. The order of the subsequent statements may not be important, but you must start with the general statement of what you want to do. You would not, for example, go up to a bank teller and say, "Give me five 20s and two 50s." This is not only bad form, but would probably make the teller's heart skip a beat or two. You must also make sure that all the subsequent statements belong with the first. You would not say, "I want the largest box you have" when making a withdrawal from your checking account. That statement belongs with "I would like to open a safe deposit box." A SAS program is an ordered set of SAS statements like the ordered set of instructions you use when you go to the bank.

SAS statements As with any language, there are a few rules to follow when writing SAS programs. Fortunately for us, the rules for writing SAS programs are much fewer and simpler than those for English.

The most important rule is

Every SAS statement ends with a semicolon.

This sounds simple enough. But while children generally outgrow the habit of forgetting the period at the end of a sentence, SAS programmers never seem to outgrow forgetting the semicolon at the end of a SAS statement. Even the most experienced SAS programmer will at least occasionally forget the semicolon. You will be two steps ahead if you remember this simple rule.

Layout of SAS programs There really aren't any rules about how to format your SAS program. While it is helpful to have a neat looking program with each statement on a line by itself and indentions to show the various parts of the program, it isn't necessary.

- ◆ SAS statements can be in upper- or lowercase.
- ◆ Statements can continue on the next line (as long as you don't split words in two).
- ◆ Statements can be on the same line as other statements.
- ◆ Statements can start in any column.

So you see, SAS is so flexible that it is possible to write programs so disorganized that no one can read them, not even you. (Of course, we don't recommend this.)

Comments To make your programs more understandable, you can insert comments into your programs. It doesn't matter what you put in your comments—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.

There are two styles of comments you can use: one starts with an asterisk (*) and ends with a semicolon (;). The other style starts with a slash asterisk (/*) and ends with an asterisk slash (*). The following SAS program shows the use of both of these style comments:

```
* Read animals' weights from file;
DATA animals;
    INFILE 'c:\MyRawData\Zoo.dat';
    INPUT Lions Tigers;
PROC PRINT DATA = animals; /* Print the results */
RUN;
```

Since some operating environments 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.

Programming tips People who are just starting to learn a programming language often get frustrated because their programs do not work correctly the first time they write them. Writing programs should be done in small steps. Don't try to tackle a long complicated program all at once. If you start small, build on what works, and always check your results along the way, you will increase your programming efficiency. Sometimes programs that do not produce errors are still incorrect. This is why it is vital to check your results as you go even when there are no errors. If you do get errors, don't worry. Most programs simply don't work the first time, if for no other reason than you are human. You forget a semicolon, misspell a word, have your fingers in the wrong place on the keyboard. It happens. Often one small mistake can generate a whole list of errors. If you build your programs piece by piece, programs are much easier to correct when something goes wrong.

1.2 SAS Data Sets

Before you run an analysis, before you write a report, before you do anything with your data, SAS must be able to read your data. Before SAS can analyze your data, the data must be in a special form called a SAS data set. (See section 2.1 for exceptions.) Getting your data into a SAS data set is usually quite simple as SAS is very flexible and can read almost any data. Once your data have been read into a SAS data set, SAS keeps track of what is where and in what form. All you have to do is specify the name and location of the data set you want, and SAS figures out what is in it.

Variables and observations Data, of course, are the primary constituent of any data set. In traditional SAS terminology the data consist of variables and observations. Adopting the terminology of relational databases, SAS data sets are also called tables, observations are also called rows, and variables are also called columns. Below you see a rectangular table containing a small data set. Each line represents one observation, while Id, Name, Height, and Weight are variables. The data point Charlie is one of the values of the variable Name and is also part of the second observation.

Variables (Also Called Columns)				
	Id	Name	Height	Weight
Observations (Also Called Rows)	1	Susie	42	41
	2	Charlie	46	55
	3	Calvin	40	35
	4	Lucy	46	52
	5	Dennis	44	.
	6		43	50

Data types Raw data come in many different forms, but SAS simplifies this. In SAS there are just two data types: numeric and character. Numeric fields are, well, numbers. They can be added and subtracted, can have any number of decimal places, and can be positive or negative. In addition to numerals, numeric fields can contain plus signs (+), minus signs (-), decimal points (.), or E for scientific notation. Character data are everything else. They may contain numerals, letters, or special characters (such as \$ or !) and can be up to 32,767 characters long.

If a variable contains letters or special characters, it must be a character variable. However, if it contains only numbers, then it may be numeric or character. You should base your decision on how you will use the variable. (If disk space is a problem, you may also choose to base your decision on storage size. See section 11.14.) Sometimes data that consist solely of numerals make more sense as character data than as numeric. ZIP codes, for example, are made up of numerals, but it just doesn't make sense to add, subtract, multiply, or divide ZIP codes. Such numbers make more sense as character data. In the previous data set, Name is obviously a character variable, and Height and Weight are numeric. Id, however, could be either numeric or character. It's your choice.

Missing data Sometimes despite your best efforts, your data may be incomplete. The value of a particular variable may be missing for some observations. In those cases, missing character data are represented by blanks, and missing numeric data are represented by a single period (.). In the preceding data set, the value of Weight for observation 5 is missing, and its place is marked by a period. The value of Name for observation 6 is missing and is just left blank.

Size of SAS data sets Prior to SAS 9.1, SAS data sets could contain up to 32,767 variables. Beginning with SAS 9.1, the maximum number of variables in a SAS data set is limited by the resources available on your computer—but SAS data sets with more than 32,767 variables cannot be used with earlier versions of SAS. The number of observations, no matter which version of SAS you are using, is limited only by your computer's capacity to handle and store them.

Rules for names of variables and SAS data set members You make up names for the variables in your data and for the data sets themselves. It is helpful to make up names that identify what the data represent, especially for variables. While the variable names A, B, and C might seem like perfectly fine, easy-to-type names when you write your program, the names Sex, Height, and Weight will probably be more helpful when you go back to look at the program six months later. Follow these simple rules when making up names for variables and data set members:

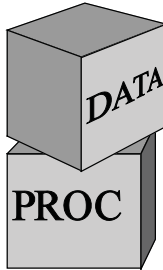
- ◆ Names must be 32 characters or fewer in length.
- ◆ Names must start with a letter or an underscore (_).
- ◆ Names can contain only letters, numerals, or underscores (_). No %\$!*&#@, please.¹
- ◆ Names can contain upper- and lowercase letters.

This last point is an important one. SAS is insensitive to case so you can use uppercase, lowercase, or mixed case—whichever looks best to you. SAS doesn't care. The data set name heightweight is the same as HEIGHTWEIGHT or HeightWeight. Likewise, the variable name BirthDate is the same as BIRTHDATE and birThDaTe. However, there is one difference for variable names. SAS remembers the case of the first occurrence of each variable name and uses that case when printing results. That is why, in this book, we use mixed case for variable names but lowercase for other SAS names.

Documentation stored in SAS data sets In addition to your actual data, SAS data sets contain information about the data set such as its name, the date that you created it, and the version of SAS you used to create it. SAS also stores information about each variable, including its name, label (if any), type (numeric or character), length (or storage size), and position within the data set. This information is sometimes called the descriptor portion of the data set, and it makes SAS data sets self-documenting.

¹ It is possible to use special characters, including spaces, in variable names if you use the system option VALIDVARNames=ANY and a name literal of the form 'variable-name'N. Starting with SAS 9.3, some special characters are allowed in SAS data set names when not running in the SAS windowing environment.

1.3 DATA and PROC Steps



SAS programs are constructed from two basic building blocks: DATA steps and PROC steps. A typical program starts with a DATA step to create a SAS data set and then passes the data to a PROC step for processing. Here is a simple program that converts miles to kilometers in a DATA step and prints the results with a PROC step:

DATA step	[	<pre>DATA distance; Miles = 26.22; Kilometers = 1.61 * Miles;</pre>
PROC step	[	<pre>PROC PRINT DATA = distance; RUN;</pre>

DATA and PROC steps are made up of statements. A step may have as few as one or as many as hundreds of statements. Most statements work in only one type of step—in DATA steps but not PROC steps, or vice versa. A common mistake made by beginners is to try to use a statement in the wrong kind of step. You're not likely to make this mistake if you remember that DATA steps read and modify data while PROC steps analyze data, perform utility functions, or print reports.

DATA steps start with the DATA statement, which starts, not surprisingly, with the word DATA. This keyword is followed by a name that you make up for a SAS data set. The DATA step above produces a SAS data set named DISTANCE. In addition to reading data from external, raw data files, DATA steps can include DO loops, IF-THEN/ELSE logic, and a large assortment of numeric and character functions. DATA steps can also combine data sets in just about any way you want, including concatenation and match-merge.

Procedures, on the other hand, start with a PROC statement in which the keyword PROC is followed by the name of the procedure (PRINT, SORT, or MEANS, for example). Most SAS procedures have only a handful of possible statements. Like following a recipe, you use basically the same statements or ingredients each time. SAS procedures do everything from simple sorting and printing to analysis of variance and 3D graphics.

A step ends when SAS encounters a new step (marked by a DATA or PROC statement); a RUN, QUIT, STOP, or ABORT statement; or, if you are running in batch mode, the end of the program. RUN statements tell SAS to run all the preceding lines of the step and are among those rare, global statements that are not part of a DATA or PROC step. In the program above, SAS knows that the DATA step has ended when it reaches the PROC statement. The PROC step ends with a RUN statement, which coincides with the end of the program.

While a typical program starts with a DATA step to input or modify data and then passes the data to a PROC step, that is certainly not the only pattern for mixing DATA and PROC steps. Just as you can stack building blocks in any order, you can arrange DATA and PROC steps in any order. A program could even contain only DATA steps or only PROC steps.

To review, the table below outlines the basic differences between DATA and PROC steps:

DATA steps	PROC steps
▶ begin with DATA statements	▶ begin with PROC statements
▶ read and modify data	▶ perform specific analysis or function
▶ create a SAS data set	▶ produce results or report

As you read this table, keep in mind that it is a simplification. Because SAS is so flexible, the differences between DATA and PROC steps are, in reality, more blurry. The table above is not meant to imply that PROC steps never create SAS data sets (most do), or that DATA steps never produce reports (they can). Nonetheless, you will find it much easier to write SAS programs if you understand the basic functions of DATA and PROC steps.

1.4 The DATA Step's Built-in Loop

DATA steps read and modify data, and they do it in a way that is flexible, giving you lots of control over what happens to your data. However, DATA steps also have an underlying structure, an implicit, built-in loop. You don't tell SAS to execute this loop: SAS does it automatically. Memorize this:

DATA steps execute line by line and observation by observation.

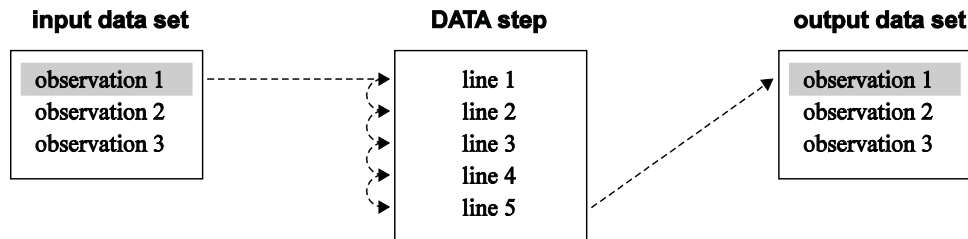
This basic concept is rarely stated explicitly. Consequently, new users often grow into old users before they figure this out on their own.

The idea that DATA steps execute line by line is fairly straightforward and easy to understand. It means that, by default, SAS executes line one of your DATA step before it executes line two, and line two before line three, and so on. That seems common sense, and yet new users frequently run into problems because they try to use a variable before they create it. If a variable named Z is the product of X and Y, then you better make sure that the statements creating X and Y come before the statements creating Z.

What is not so obvious is that while DATA steps execute line by line, they also execute observation by observation. That means SAS takes the first observation and runs it all the way through the DATA step (line by line, of course) before looping back to pick up the second observation. In this way, SAS sees only one observation at a time.

Imagine a SAS program running in slow motion: SAS reads observation number one from your input data set. Then SAS executes your DATA step using that observation. If SAS reaches the end of the DATA step without encountering any serious errors, then SAS writes the current observation to a new, output data set and returns to the beginning of the DATA step to process the next observation. After the last observation has been written to the output data set, SAS terminates the DATA step and moves on to the next step, if there is one. End of slow motion; please return to normal gigahertz.

This diagram illustrates how an observation flows through a DATA step:



SAS reads observation number one and processes it using line one of the DATA step, then line two, and so on until SAS reaches the end of the DATA step. Then SAS writes the observation in the output data set. This diagram shows the first execution of the line-by-line loop. Once SAS finishes with the first observation, it loops back to the top of the DATA step and picks up observation two. When SAS reaches the last observation, it automatically stops.

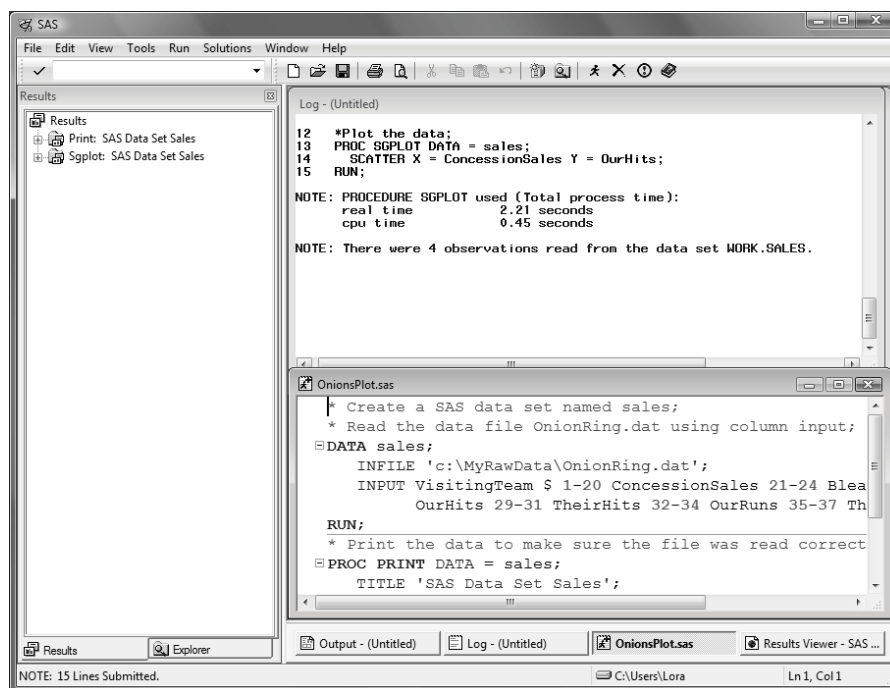
Here is an analogy. DATA step processing is a bit like voting. When you arrive at your polling place, you stand in line behind other people who have come to vote. When you reach the front of the line you are asked standard questions: “What is your name? Where do you live?” Then you sign your name, and you cast your vote. In this analogy, the people are observations, and the voting process is the DATA step. People vote one at a time (or observation by observation). Each voter’s choices are secret, and peeking at your neighbor’s ballot is definitely frowned upon. In addition, each person completes each step of the process in the same order (line by line). You cannot cast your vote before you give your name and address. Everything must be done in the proper order.

If this seems a bit too structured, SAS offers a number of ways to override the line-by-line and observation-by-observation structure. These include the RETAIN statement (discussed in section 3.10) and the OUTPUT statement (discussed in sections 6.9 and 6.10).

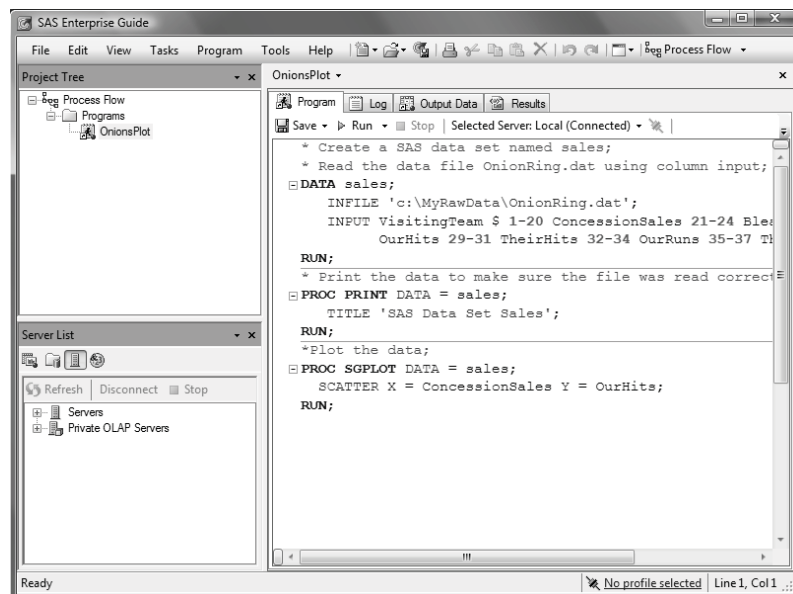
1.5 Choosing a Mode for Submitting SAS Programs

So far we have talked about writing SAS programs, but simply writing a program does not give you any results. Just like writing a letter to your representative in Congress does no good unless you mail it, a SAS program does nothing until you submit or execute it. You can execute a SAS program several ways, but not all methods are available for all operating environments. Check the SAS Help and Documentation for your operating environment to find out which methods are available to you. The method you choose for executing a SAS program will depend on your preferences and on what is most appropriate for your application and your environment. If you are using SAS at a large site with many users, then ask around and find out which is the most accepted method of executing SAS. If you are using SAS on your own personal computer, then choose the method that suits you.

SAS windowing environment If you type SAS at your system prompt, or click the SAS icon, you will most likely get into the SAS windowing environment (also known as Display Manager). In this interactive environment, you can write and edit SAS programs, submit programs for processing, and view and print your results. In addition, there are many SAS windows for performing different tasks such as managing SAS files, customizing the interface, accessing SAS Help and Documentation, and importing or exporting data. Exactly what your windowing environment looks like depends on the type of computer you are using, the operating environment on the computer, and what options are in effect when you start up SAS. If you are using a personal computer, then the SAS windowing environment will look similar to other programs on your computer, and many of the features will be familiar to you. The following figure shows the SAS windowing environment in Microsoft Windows.



SAS Enterprise Guide If you have SAS Enterprise Guide software, which runs only under Windows, you may choose to submit your programs from within SAS Enterprise Guide. To do this, open a Program window where you can type in your SAS program or open an existing SAS program. The program editor in SAS Enterprise Guide (starting with version 4.3) displays automatic syntax help as you type your program, and there is a program analyzer that will generate a diagram of your program to help you visualize the parts and how they fit together. You can choose to run your code on the local machine, or on a remote server where SAS is installed. To run your SAS program on a remote server, you may need to have additional SAS software installed. Also, SAS Enterprise Guide can write SAS code for you through its extensive menu system. SAS Enterprise Guide is project based, so all your programs, results, and references to data are stored in one project file. The following figure shows a project in SAS Enterprise Guide 4.3.



Batch or background mode With batch or background mode, your SAS program is in a file. You submit the file for processing with SAS. Your SAS program may start executing immediately, or it could be put in a queue behind other jobs. Batch processing is used a lot on mainframe computers. You can continue to work on your computer while your job is being processed, or better yet, you can go to the baseball game and let the computer work in your absence. Batch processing is usually less expensive than other methods and is especially good for large jobs which can be set up to execute during off-hours when the rates are at their lowest. When your job is complete, the results will be placed in a file or files, which you can display or print at any time.

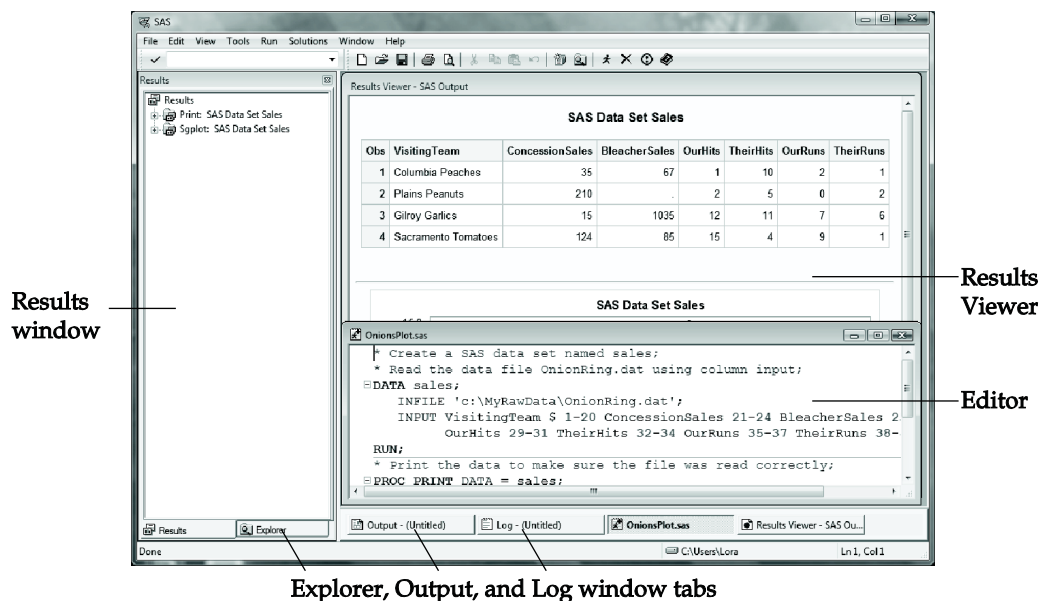
To find out how to submit SAS programs for batch processing, check the SAS Help and Documentation for your operating environment, or check with other SAS users at your site. Even sites with the same operating environment may have different ways of submitting jobs in batch mode.

1.6 Windows and Commands in the SAS Windowing Environment

The SAS windowing environment (also known as Display Manager) adopts the look and feel of your operating environment. This is good for you because many aspects of the SAS windowing environment will be familiar. But there are many ways in which you can customize your SAS environment if you want. This makes writing about it challenging, because we can't tell you exactly what your SAS session will look like and how it will behave. However, there are common elements between the various operating environments, and you will probably already be familiar with those elements which are different.

The SAS Windows

There are five basic SAS windows: the Results and Explorer windows, and three programming windows: Editor, Log, and Output. In the Windows operating environment a sixth window, the Results Viewer appears if you run a program that generates printable results. Sometimes the windows are not immediately visible. For example, in the Windows operating environment, the Output window initially appears behind the Editor and Log windows. There are also many other SAS windows that you may use for tasks such as getting help, changing SAS system options, and customizing your SAS session. The following figure shows the windows for a Microsoft Windows SAS session, with pointers to the main SAS windows.



Editor This window is a text editor. You can use it to type in, edit, and submit SAS programs as well as edit other text files such as raw data files. In Windows operating environments, the default editor is the Enhanced Editor. The Enhanced Editor is syntax sensitive and color codes your programs making it easier to read them and find mistakes. The Enhanced Editor also allows you to collapse and expand the various steps in your program. For other operating environments, the default editor is the Program Editor whose features vary with the version of SAS and operating environment.

Log The Log window contains notes about your SAS session, and after you submit a SAS program, any notes, errors, or warnings associated with your program as well as the program statements themselves will appear in the Log window.

Output In the z/OS operating environment, all tabular results will appear in the Output window. By default, in the Windows and UNIX environments, nothing appears in the Output window. But if you turn on the LISTING destination (see section 5.4), then results will appear in the Output window.

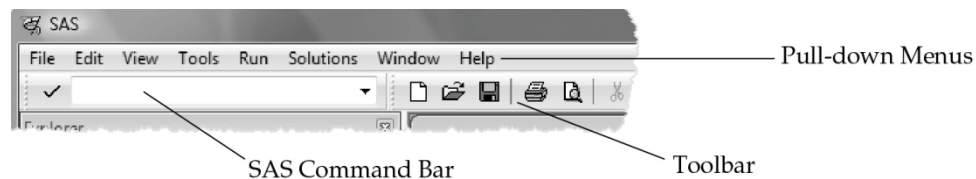
Results Viewer In the Windows operating environment, if your program generates any printable results, then the Results Viewer window will open and display the results.

Results The Results window is like a table of contents for your Output and Results Viewer windows; the results tree lists each part of your results in an outline form.

Explorer The Explorer window gives you easy access to your SAS files and libraries.

The SAS Commands

There are SAS commands for performing a variety of tasks. Some tasks are probably familiar, such as opening and saving files, cutting and pasting text, and accessing Help. Other commands are specific to the SAS System, such as submitting a SAS program. You may have up to three ways to issue commands: menus, the toolbar, or the SAS command bar (or command line). The following figure shows the location of these three methods of issuing SAS commands in the Windows operating environment default view.



Menus Most operating environments will have pull-down menus located either at the top of each window, or at the top of your screen. If your menus are at the top of your screen, then the menus will change when you activate the different windows (usually by clicking on them). You may also have, for each window, context-sensitive pop-up menus that appear when you click the right or center button of your mouse.

Toolbar The toolbar, if you have one, gives you quick access to commands that are already accessible through the pull-down menus. Not all operating environments have a toolbar.

SAS command bar The command bar is a place that you can type in SAS commands. In some operating environments the command bar is located with the toolbar (as shown here); in other operating environments you may have a command line with each of the SAS windows (usually indicated by `Command=>`). Most of the commands that you can type in the command bar are also accessible through the pull-down menus or the toolbar.

Controlling your windows The Window pull-down menu gives you choices on how the windows are placed on your screen. You can also activate any of the programming windows by selecting it from the Window pull-down menu, or by simply clicking the window.

1.7 Submitting a Program in the SAS Windowing Environment

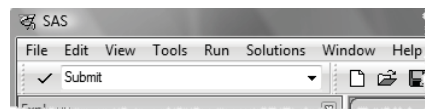
Naturally after going to the trouble of writing SAS programs, you want to see some results. As we have already discussed, there are several ways of submitting SAS programs. If you use the SAS windowing environment, then you can edit and submit programs, and see results all within the windowing environment.

Getting your program into the editor The first thing you need to do is get your program into the Editor window. You can either type your program into the editor, or you can bring the program into the Editor window from a file. The commands for editing in the editor and for opening files should be familiar. SAS tries to follow conventions that are common for your operating environment. For example, to open a file in the editor, you can select **File ► Open** from the menu bar. For some operating environments you may have an Open icon on the toolbar, and you may also have the option of pasting your file into the editor from the clipboard.

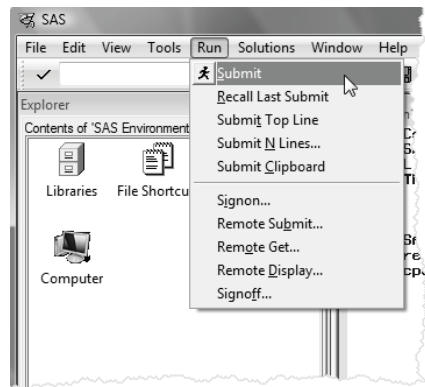
Submitting your program Once your program appears in the editor, you execute it (either the whole program or a part you have highlighted) using the SUBMIT command. Depending on your operating environment, you have a few choices on how to execute the SUBMIT command. First click the Editor window to make it active. Then do one of the following:



Click the **Submit** button on the toolbar.



Enter SUBMIT in the command line area of your SAS session.

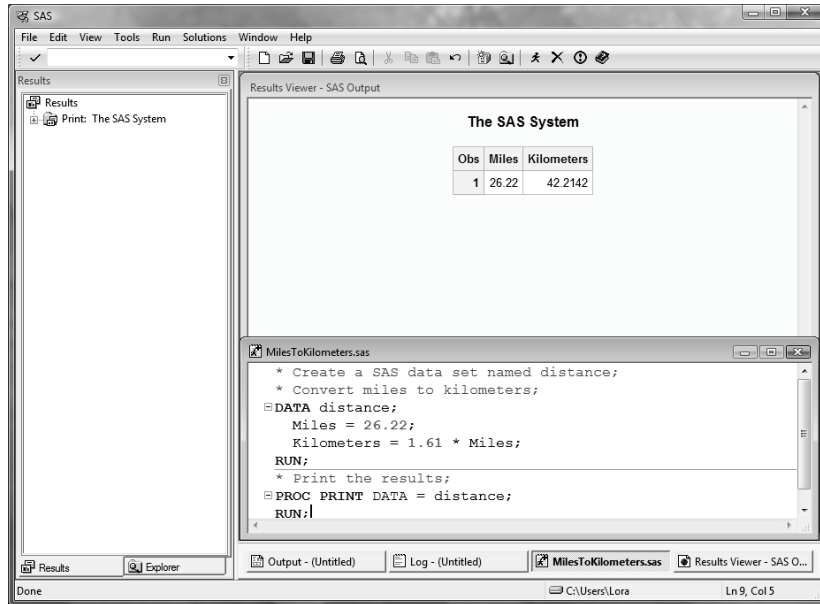


Select **Run ► Submit** from the menu bar.

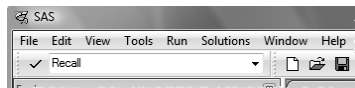
Viewing the SAS Log and Results In the Windows operating environment, after you submit your program, the program remains in the Enhanced Editor window and the results of your program go into the Log and Results Viewer windows. In the UNIX environment, your results go into the Log window and a separate web browser window, while in z/OS your results go into the Log and Output windows. For both UNIX and z/OS, after submitting your program it disappears from the Program Editor window. At first it may be a shock for you to see your program disappear in front of your eyes.

Don't worry; the program you spent so long writing is not gone forever. If your program produced any output, then you will also get new entries in the Results window. The Results window is like a table of contents for your SAS output and is discussed in more detail in section 1.9. The following figure is an example of what your screen might look like after you submit a program from the Enhanced Editor in the Windows environment.

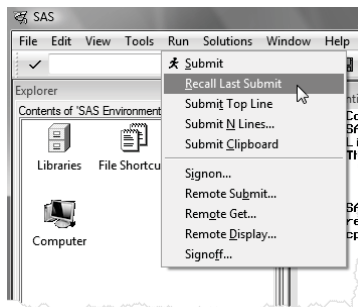
You may not see all the windows at the same time. In some operating environments, the windows are placed one on top of the other. In this figure the Explorer window is under the Results window and the Output and Log windows are beneath the Enhanced Editor and Results Viewer windows. You can bring a window to the top by clicking it or its tab, typing its name in the command line area, or selecting it from the Window menu.



Getting your program back Unfortunately for most of us, our programs do not run perfectly every time. If you have an error in your program, you will most likely want to edit the program and run it again. If you are using the Enhanced Editor, then your program will remain in the window after you submit it. However, if you are using the Program Editor window, you will need to get your program back in the Program Editor window using the RECALL command. You have two choices for executing the RECALL command.



Make the Program Editor the active window, then enter RECALL in the command line area of your SAS session.



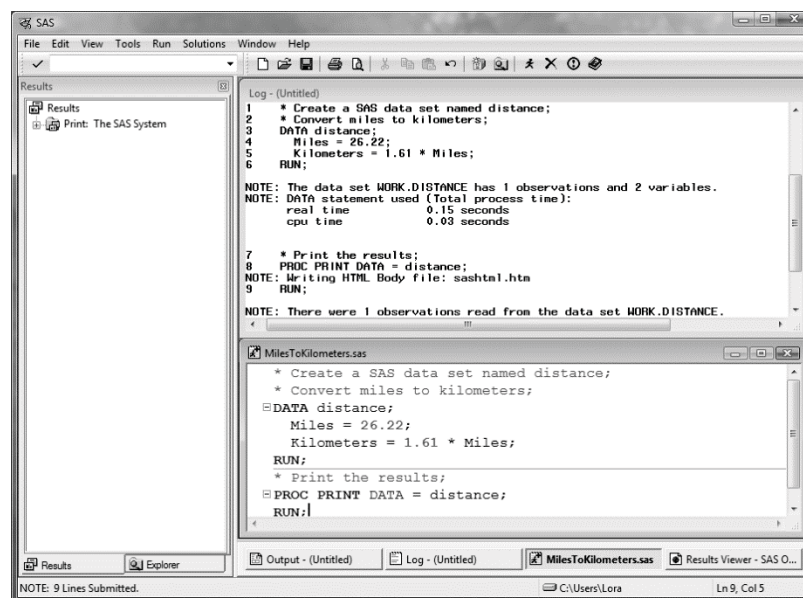
Make the Program Editor the active window, then select **Run ► Recall Last Submit** from the menu bar.

The RECALL command will bring back the last block of statements you submitted. If you use the RECALL command again, it will insert the block of statements submitted before the last one, and so on and so on, until it retrieves all the statements you submitted.

1.8 Reading the SAS Log

Every time you submit a SAS program, SAS writes messages in your log. Many SAS programmers ignore the SAS log and go straight to the output. That's understandable, but dangerous. It is possible—and sooner or later it happens to all of us—to get bogus results that look fine in the output. The only way to know they are bad is to check the SAS log. Just because it runs doesn't mean it's right.

Where to find the SAS log The location of the SAS log varies depending on the operating environment you use, the mode you use (SAS windowing environment or batch), and local settings. If you submit a program in the windowing environment, you will, by default, see the SAS log in your Log window as in the following figure.



If you submit your program in batch mode, the log will be written to a file that you can view or print using your operating environment's commands for viewing and printing. The name given to the log file is generally some permutation of the name you gave the original program. For example, if you named your SAS program

Marathon.sas, then it is a good bet that your log file will be Marathon.log.

What the log contains People tend to think of the SAS log as either a rehash of their program or as just a lot of gibberish. OK, we admit, there is some technical trivia in the SAS log, but there is also plenty of important information. Here is a simple program that converts miles to kilometers and prints the result:

```

* Create a SAS data set named distance;
* Convert miles to kilometers;
DATA distance;
    Miles = 26.22;
    Kilometers = 1.61 * Miles;
RUN;
* Print the results;
PROC PRINT DATA = distance;
RUN;

```

If you run this program, SAS will produce a log similar to this:

```

❶ NOTE: Copyright (c) 2002-2010 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Version 9.3 (TS1M0)
      Licensed to XYZ Inc., Site 0099999001.
NOTE: This session is executing on the W32_VSPRO platform.
NOTE: SAS initialization used:
      real time          1.40 seconds
      cpu time           0.96 seconds

❷ 1  * Create a SAS data set named distance;
   2  * Convert miles to kilometers;
   3  DATA distance;
   4      Miles = 26.22;
   5      Kilometers = 1.61 * Miles;
   6  RUN;

❸ NOTE: The data set WORK.DISTANCE has 1 observations and 2 variables.

❹ NOTE: DATA statement used (Total process time):
      real time          0.03 seconds
      cpu time           0.03 seconds

❷ 7  * Print the results;
   8  PROC PRINT DATA = distance;
   9  RUN;
NOTE: There were 1 observations read from the data set WORK.DISTANCE

❺ NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.01 seconds
      cpu time           0.00 seconds

```

The SAS log above is a blow-by-blow account of how SAS executes the program.

- ❶ It starts with notes about the version of SAS and your SAS site number.
- ❷ It contains the original program statements with line numbers added on the left.
- ❸ The DATA step is followed by a note containing the name of the SAS data set created (WORK.DISTANCE), and the number of observations (1) and variables (2). A quick glance is enough to assure you that you did not lose any observations or accidentally create a lot of unwanted variables.
- ❹ Both DATA and PROC steps produce a note about the computer resources used. At first you probably won't care in the least. But if you run on a multi-user system or have long jobs with large data sets, these statistics may start to pique your interest. If you ever find yourself wondering why your job takes so long to run, a glance at the SAS log will tell you which steps are the culprits.

If there were error messages, they would appear in the log, indicating where SAS got confused and what action it took. You may also find warnings and other types of notes which sometimes indicate errors and other times just provide useful information. Chapter 11 discusses several of the more common errors SAS users encounter.

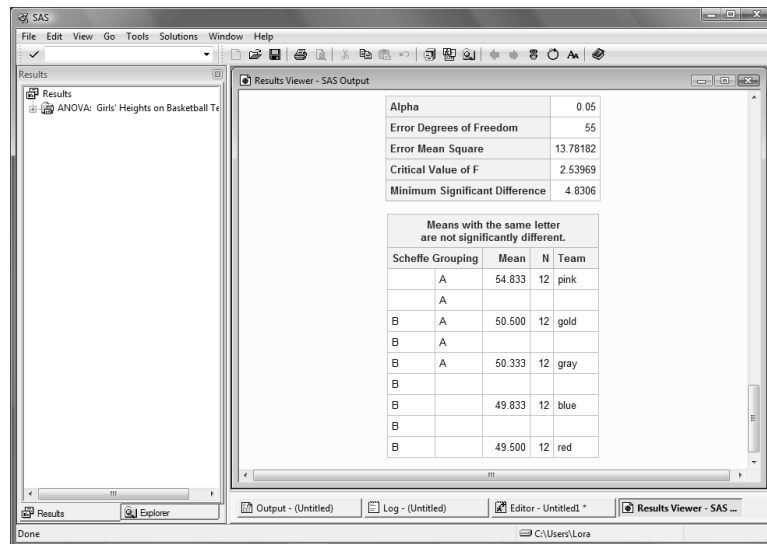
1.9 Viewing Your Results

How you view your output depends on what operating environment you are using and how you submit your program.

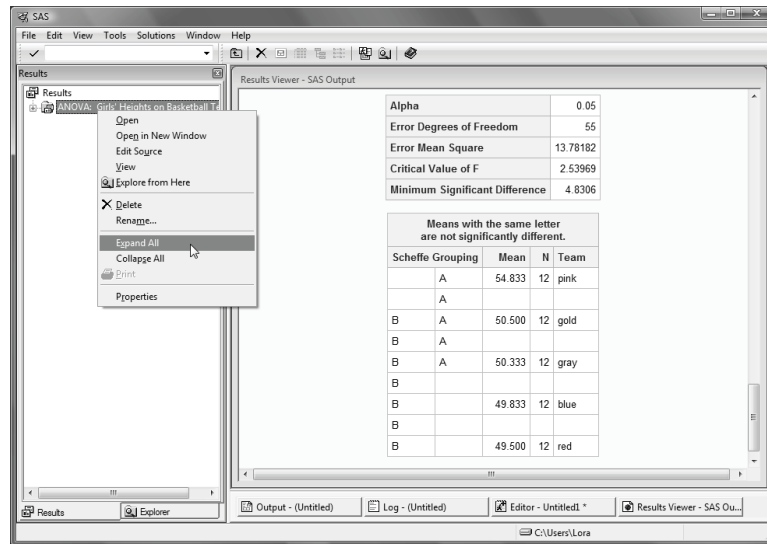
SAS windowing environment If you submit your program in the SAS windowing environment under Microsoft Windows, then your output will, by default, go to the Results Viewer window and be displayed in HTML. Under UNIX, your output will also be displayed in HTML by default, but it will be displayed in a separate web browser window. Under z/OS, output will display as text in the Output window.

Batch mode If you submit your program in batch mode, then your output will be in a file on your computer and you would use your operating environment's commands to view the output file (also called the listing). For example, if you execute your SAS program in batch mode on a UNIX system, then your output will be in a file with an extension .lst. To view the file, you can use either the `cat` or `more` commands.

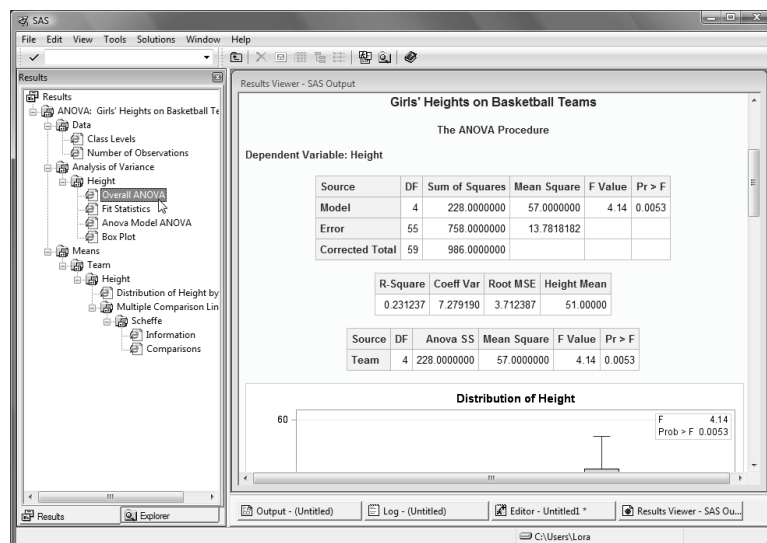
Results Viewer window After submitting your program in the SAS windowing environment under Microsoft Windows, your results will go to the Results Viewer window. The Results Viewer window is automatically opened and appears on top of any of the other programming windows that are open. The following figure shows what your Results Viewer window might look like after submitting a program containing an ANOVA (Analysis of Variance) procedure. Notice that the Results Viewer window is automatically scrolled down so that you see the end of the procedure's output.



Results window When you have a lot of output, the Results window can be very helpful. The Results window is like a table of contents for your output. It lists each procedure that produces output, and if you open or expand the procedure in the Results tree, you can see each part of the procedure output. Expand the results tree, by clicking the plus (+) signs, or by right-clicking the result and selecting **Expand All**.



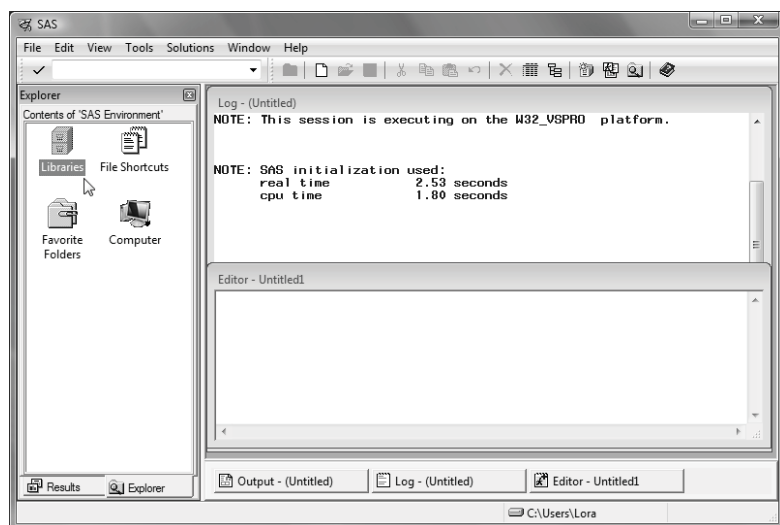
Double-click the output you want to see, and it will appear at the top of the Results Viewer window. The following figure shows what your Results Viewer window would look like after double-clicking the **Overall ANOVA** item in the Results Viewer window.



1.10 SAS Data Libraries

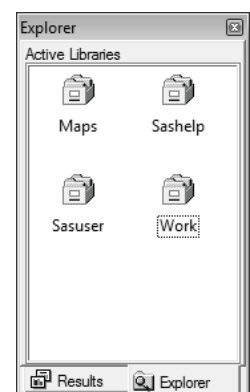
Before you can use a SAS data set, you have to tell SAS where to find it. You do that by setting up a SAS library. A SAS library is simply a location where SAS data sets (as well as other types of SAS files) are stored. Depending on your operating environment, a SAS library might be a folder or directory on your computer, or it might be a physical location like a hard drive, flash drive, or CD. To set up a library, all you have to do is make up a name for your library and tell SAS where it is. There are several ways to do this including using the LIBNAME statement (covered in sections 2.18 to 2.19) and using the New Library window in the SAS windowing environment.

When you start the SAS windowing environment, you see the basic SAS windows including the SAS Explorer window on the left. (If the Explorer window is under the Results window, click its tab to bring it forward.) If you double-click the Libraries icon, Explorer will open the Active Libraries window showing all the libraries that are currently defined. To go back to the previous window within Explorer, choose **View ► Up one level** from the menu bar, or click

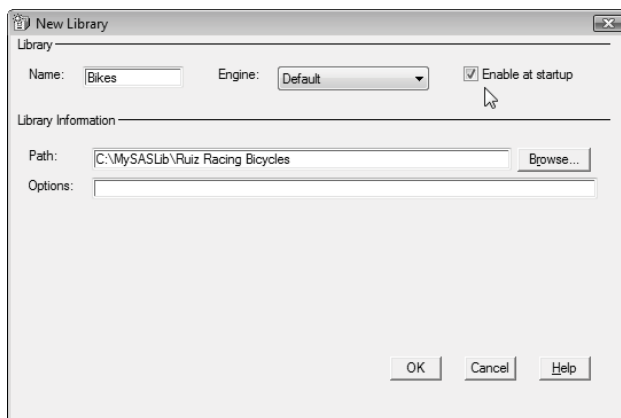
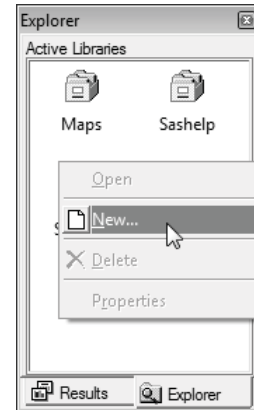


the previous window within Explorer, choose **View ► Up one level** from the menu bar, or click the Explorer window to make it active, and then click the Up One Level button  on the toolbar.

Active Libraries window When you open the Active Libraries window, you will see at least three libraries: SASHELP, SASUSER, and WORK. You may have other libraries for specific SAS products (such as the MAPS library for SAS/GRAPH software), or libraries that have been set up by you or someone you work with. The SASHELP library contains information that controls your SAS session along with sample SAS data sets. The WORK library is a temporary storage location for SAS data sets. It is also the default library. If you create a SAS data set without specifying a library, SAS will put it in the WORK library, and then delete it when you end your session. If you make changes to the default settings for the SAS windowing environment, this information will be stored in the SASUSER library. You can also store SAS data sets, SAS programs, and other SAS files in the SASUSER library. However, many people prefer to create a new library for their SAS files.



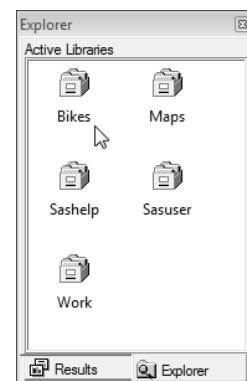
Creating a new library You can create new SAS libraries using the New Library window. To open this window, either select **Tools ► New Library** from the menu bar, or right-click the Active Libraries window and choose **New** from the pop-up menu.



In the New Library window, type the name of the library you want to create. This name is called a libref, which is short for library reference. A libref must be 8 characters or fewer; start with a letter or underscore; and contain only letters, numerals, or underscores. In this window, the name BIKES has been typed in as the libref. In the Path field, enter the complete path to the folder or directory where you want your data sets to be stored, or click the **Browse...** button to navigate to the location. If you don't want to define your library

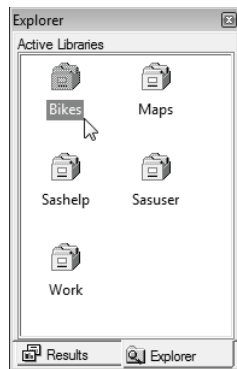
reference every time you start up SAS, then check the **Enable at startup** box. Click **OK** and then your new library reference will appear in the Active Libraries window.

Here is the Active Libraries window showing the newly created BIKES library.



1.11 Viewing Data Sets in the Viewtable Window

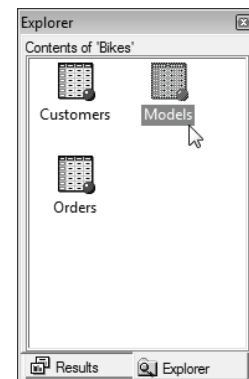
In addition to listing your current libraries and creating new libraries, you can also use SAS Explorer to open SAS data sets for viewing in Viewtable. When you are writing programs, it is always a good idea to check the data sets you create to make sure they are correct. Viewtable is one way you can look at your SAS data sets.



Start by double-clicking the Libraries icon in the Explorer window as shown in the previous section. This will open the Active Libraries window showing all the libraries that are currently defined on your system. If you double-click a library icon, SAS will open a Contents window showing you all the SAS files in that particular library.

To go back to the previous window within Explorer, choose **View ► Up one level** from the menu bar, or click the Explorer window to make it active, and then click the Up One Level button  on the toolbar.

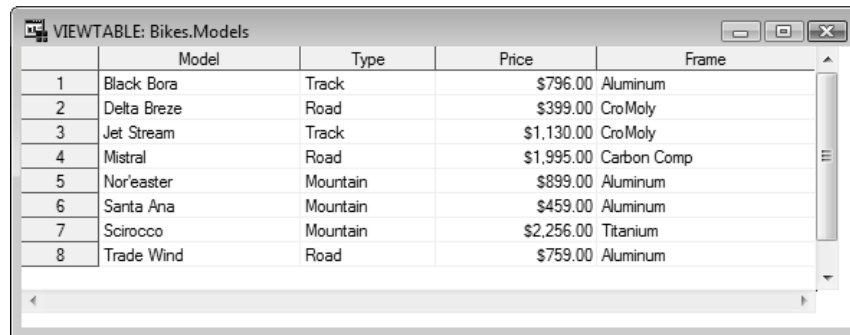
Contents window This window shows the contents of a library. SAS data sets are represented by an icon showing a little table of data and a red ball. The library shown on the right contains three data sets named CUSTOMERS, MODELS, and ORDERS. If you double-click a data set, SAS will open a Viewtable window showing that data set. (If you don't yet have any SAS data sets of your own, you can view sample data sets that are provided with SAS in the SASHELP library. The CLASS data set in the SASHELP library is a good one to view.)



Viewtable window This window allows you to create, browse, and edit data sets. When you first open SAS data sets, the data are in browse mode so you cannot make any changes. To switch to edit mode, select **Edit ► Edit Mode** from the menu bar. Creating and editing data sets using Viewtable is discussed in more detail in section 2.2. This picture shows the data set named MODELS from the BIKES library.

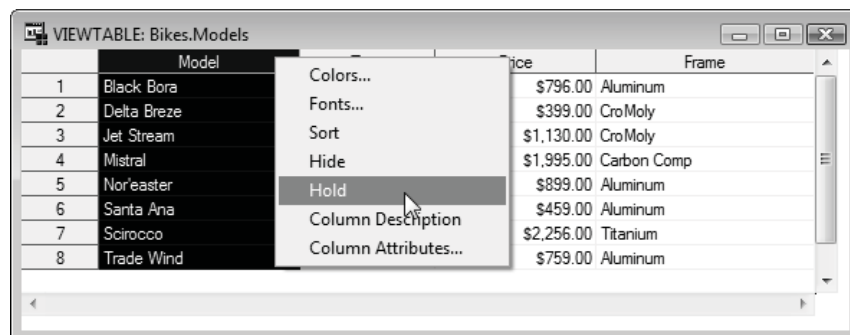
	Model Name	Type of Bicycle	List Price	Frame Composition
1	Black Bora	Track	\$796.00	Aluminum
2	Delta Breze	Road	\$399.00	CroMoly
3	Jet Stream	Track	\$1,130.00	CroMoly
4	Mistral	Road	\$1,995.00	Carbon Comp
5	Nor'easter	Mountain	\$899.00	Aluminum
6	Santa Ana	Mountain	\$459.00	Aluminum
7	Scirocco	Mountain	\$2,256.00	Titanium
8	Trade Wind	Road	\$759.00	Aluminum

Changing column headings By default, Viewtable uses variable labels for column headings, or, if a variable does not have a label, the variable name is displayed. Sometimes you may want to see the actual variable names instead of the labels. To do this, click the Viewtable window to make it active, then select **View ► Column Names** from the menu bar. Here is the MODELS SAS data set showing the column (also called variable) names instead of the labels.



	Model	Type	Price	Frame
1	Black Bora	Track	\$796.00	Aluminum
2	Delta Breze	Road	\$399.00	CroMoly
3	Jet Stream	Track	\$1,130.00	CroMoly
4	Mistral	Road	\$1,995.00	Carbon Comp
5	Nor'easter	Mountain	\$899.00	Aluminum
6	Santa Ana	Mountain	\$459.00	Aluminum
7	Scirocco	Mountain	\$2,256.00	Titanium
8	Trade Wind	Road	\$759.00	Aluminum

Column options If you right-click a column heading, several options will appear in the pop-up menu. You can control colors, fonts, and view the column attributes. You can choose to sort the data by the values in the column. If you are not in edit mode, then you are given the option of creating a new data set containing the sorted data. You can also hide or hold columns. If you choose to hide a column, the data will not be visible in the current Viewtable session. To unhide a column, select **Data ► Hide/Unhide** from the menu bar to open the Hide/Unhide window. In this window you can change the visibility of all columns. When you choose to hold a column, it and every column to the left of it will always be visible, even when you scroll to the right.

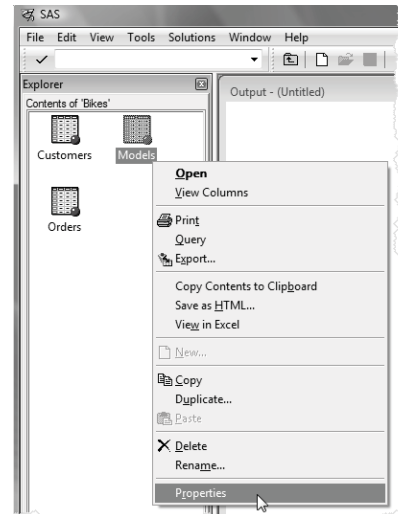
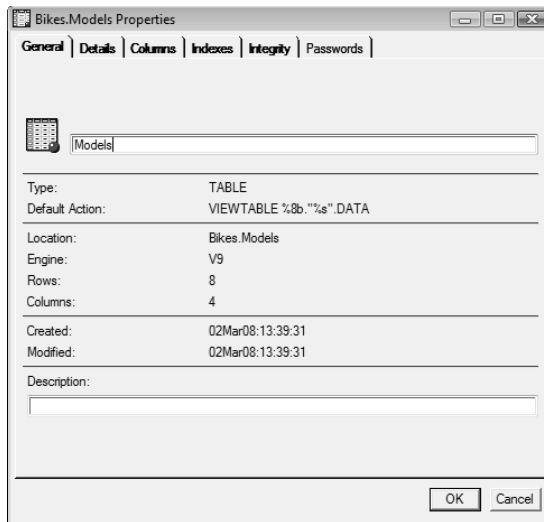


	Model	Type	Price	Frame
1	Black Bora	Track	\$796.00	Aluminum
2	Delta Breze	Road	\$399.00	CroMoly
3	Jet Stream	Track	\$1,130.00	CroMoly
4	Mistral	Road	\$1,995.00	Carbon Comp
5	Nor'easter	Mountain	\$899.00	Aluminum
6	Santa Ana	Mountain	\$459.00	Aluminum
7	Scirocco	Mountain	\$2,256.00	Titanium
8	Trade Wind	Road	\$759.00	Aluminum

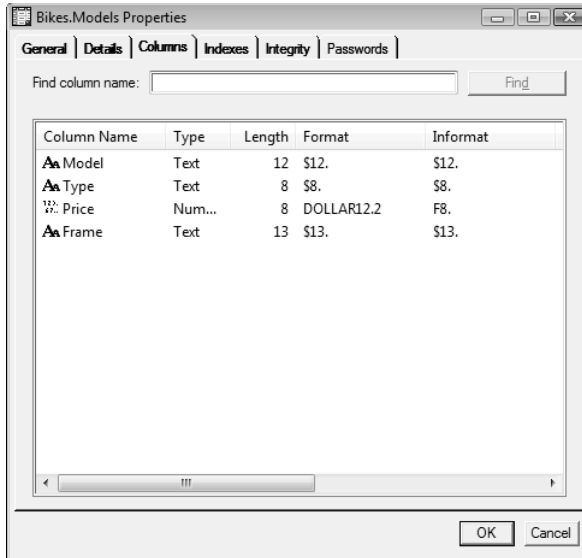
1.12 Viewing the Properties of Data Sets with SAS Explorer

The Properties window for a SAS data set contains some very useful information, such as the date and time the data set was created, the number of observations, all the variable names, and the attributes of the variables. The Properties window contains information similar to the output produced by the CONTENTS procedure described in section 2.21.

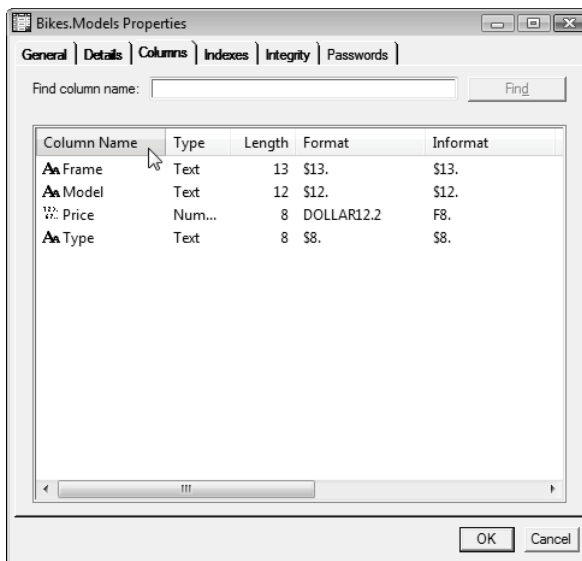
Opening the Properties window To open the Properties window, start by double-clicking the Libraries icon in the Explorer window and then double-clicking the library containing the SAS data set. SAS will display the contents of the library in the Explorer window. Right-click the icon for the data set, and select **Properties** from the pop-up menu. This opens the Properties window with the General tab on top. This figure shows what the Properties window looks like in the Microsoft Windows operating environment.



General tab This window displays information about the data set such as the date it was created and the number of rows (or observations) and columns (or variables).



Columns tab If you click the Columns tab, SAS displays information about the columns (or variables) in that data set. The variable name, type, and length are displayed along with any formats or informats assigned to the variable. The variable labels are also displayed in this window, but to see them you need to scroll to the right.



If you have lots of variables in your data set, using the sort and find features can make your work easier. You can sort any of these columns alphabetically by clicking the column heading. This window shows the variables sorted by name. You can find a column by typing its name in the box labeled **Find column name**.

1.13 Using SAS System Options

System options are parameters you can change that affect SAS—how it works, what the output looks like, how much memory is used, error handling, and a host of other things. SAS makes many assumptions about how you want it to work. This is good. You do not want to specify every little detail each time you use SAS. However, you may not always like the assumptions SAS makes. System options give you a way to change some of these assumptions.

Not all options are available for all operating environments. A list of options specific to your operating environment appears in the SAS Help and Documentation. You can see a list of system options and their current values by opening the SAS System Options window, or by using the `OPTIONS` procedure. To use the `OPTIONS` procedure, submit the following SAS program and view the results in the SAS log:

```
PROC OPTIONS;
RUN;
```

There are four ways to specify system options. Some options can be specified using only some of these methods. The SAS Help and Documentation for your operating environment tells you which methods are valid for each system option:

1. Create a SAS configuration file which contains settings for the system options. This file is accessed by SAS every time SAS is started. Configuration files are created by systems administrators. (This could be you if you are using a PC.)
2. Specify system options at the time you start up SAS from your system's prompt (called the invocation).
3. Change selected options in the SAS System Options window if you are using the SAS windowing environment.
4. Use the `OPTIONS` statement as a part of your SAS program.

The methods are listed here in order of increasing precedence; method 2 will override method 1, method 3 will override method 2, and so forth. If you are using the SAS windowing environment, methods 3 and 4, the SAS System Options window and `OPTIONS` statement, will override each other—so whichever was used last will be in effect. Only the last two methods are covered here. The first two methods are very system dependent; to find out more about these methods see the SAS Help and Documentation for your operating environment.

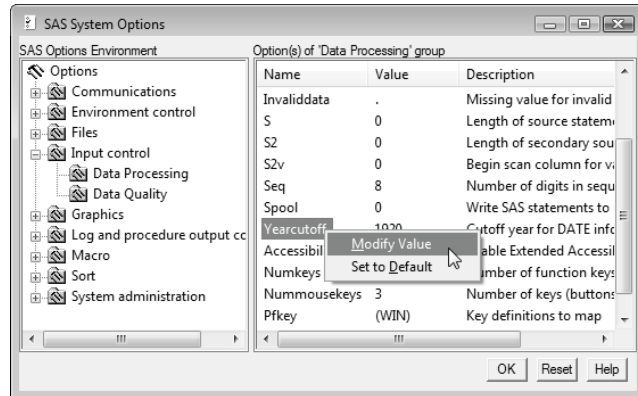
OPTIONS statement The `OPTIONS` statement is part of a SAS program and affects all steps that follow it. It starts with the keyword `OPTIONS` and follows with a list of options and their values. For example

```
OPTIONS LEFTMARGIN = 1IN NODATE;
```

The `OPTIONS` statement is one of the special SAS statements which do not belong to either a `PROC` or a `DATA` step. This global statement can appear anywhere in your SAS program, but it usually makes the most sense to let it be the first line in your program. This way you can easily see which options are in effect. If the `OPTIONS` statement is in a `DATA` or `PROC` step, then it affects that step and the following steps. Any subsequent `OPTIONS` statements in a program override previous ones.

SAS System Options window

You can view and change SAS system options through the SAS System Options window. Open it either by typing `OPTIONS` in the command line area on your screen, or by selecting **Tools ► Options ► System** from the menu bar. To change the value of an option, first locate the option by clicking the appropriate category on the left side of the screen. A list of options and their current values will appear on the right side of the screen. Right-click the option itself to modify the value or set it to the default.



Options for printed results The following are some system options you might want to use that affect the appearance of results in formats meant for printing (in other words not HTML):

`CENTER | NOCENTER`

controls whether output is centered or left-justified. Default is `CENTER`.

`DATE | NODATE`

controls whether or not today's date will appear at the top of each page of output. Default is `DATE`.

`NUMBER | NONUMBER`

controls whether or not page numbers appear on each page of SAS output. Default is `NUMBER`.

`ORIENTATION = orientation`

specifies the orientation for printing output, either `LANDSCAPE` or `PORTRAIT`. Default is `PORTRAIT`.

`PAGENO = n`

starts numbering output pages with *n*. Default is 1.

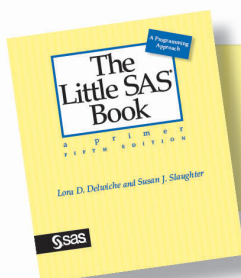
`RIGHTMARGIN = n`

specifies the size of the margin (such as 0.75in or 2cm) to be used for printing output. Default is 0.00in.

`LEFTMARGIN = n`

`TOPMARGIN = n`

`BOTTOMMARGIN = n`



From *The Little SAS® Book, Fifth Edition*.
Full book available for purchase [here](#).

Index

A

- ACROSS usage option 136, 138-139
- ACROSS= option in KEYLEGEND statement 244
- Active Libraries window 20-22
- AFTER location in REPORT procedure 140-141
- age, calculating 88-89
- AGREE option in FREQ procedure 264, 266
- AGREEMENTPLOT option in TTEST procedure 262
- AGREEPLOT option in FREQ procedure 266
- ALL keyword in TABULATE procedure 126
- ALL option in TTEST procedure 262
- ALPHA= option
 - fitted curves 240
 - MEANS procedure 258-259
 - TTEST procedure 260
 - VBAR or HBAR statement 230
- analysis of variance 276-279
- ANALYSIS usage option 136-137
- AND operator 82-83, 102
- annotation in graphics 244-245
- ANOVA procedure 276-279
- ANYALNUM function 78-79
- ANYALPHA function 78-79
- ANYDIGIT function 78-79
- ANYDTDTEw. informat 44-45, 90-91
- ANYSPACE function 78-79
- arithmetic operators 74-75
- ARRAY statement 94-95
- arrays
 - SAS arrays 94-95
- ASCII files 36
- ASCII sort order 106
- assignment statements 74-75
 - dates 88-89
 - functions 76-79
- ATTRIB statement 318
- attributes, style
 - PRINT procedure 166-167

- REPORT procedure 168-169
 - table of 174-175
- TABULATE procedure 170-171
- autocall library, macro 217
- automatic variables
 - _ERROR_ 206
 - _N_ 206-207
 - FIRST.byvariable 206-207
 - LAST.byvariable 206-207
 - macro 214-215
- axes, controlling in graphs 242-243
- AXIS= option in REFLINE statement 242

B

- BACKGROUND style attribute 174-175
- BACKGROUNDIMAGE style attribute 174-175
- bar charts 230-231, 266-267
- BARWIDTH= option in VBAR or HBAR statement 230
- batch mode 11
- BCOLOR= option in TITLE statement 164-165
- BEFORE location in REPORT procedure 140
- BESTw. format 112-113
- BETA option for distribution plots 256
- BETWEEN AND operator 102
- BINSTART= option in HISTOGRAM statement 232
- BINWIDTH= option in HISTOGRAM statement 232-233
- BMP image format 250-251
- BODY= option in ODS HTML statement 158-159
- BODYTITLE option in ODS RTF statement 160-161
- BOLD option in TITLE statement 164-165
- bolding in graphics 246-247
- BON option in ANOVA procedure 276
- Bonferroni t tests 276
- BORDER option in INSET option 244
- BOTTOMMARGIN= system option 27
- Bowker's test 264
- box plots 234-235, 276-277
- BOX= option in TABULATE procedure 128-129
- BOXPLOT option in TTEST procedure 262

BREAK statement in REPORT procedure
140-141

BY groups, definition 104

BY statement 100

 compared to PANELBY 249

 FIRST.byvariable 206-207

 ID statement with BY 189

 LAST.byvariable 206-207

 MEANS procedure 118-119

 MERGE statement 184-187

 PRINT procedure 108-109

 SET statement 182-183

 SORT procedure 104-107

 TRANPOSE procedure 204-205

 UPDATE statement 192-193

BY variables

 definition 104

 FIRST. and LAST. 206-207

 sorting 104-107

C

CALL SYMPUT 222-223

capitalization in SAS programs xiii, 5

CARDS statement 36

CAT function 78-79

CATEGORY= option

 HBOX or VBOX statement 234-235

CATS function 78-79

CATX function 78-79

CDFPLOT statement in UNIVARIATE procedure
256

CENTER system option 27

CHAR option in REPORT procedure 144-145

character data

 converting to numeric 310-311

 definition 4

 formats 112-113

 functions 78-79

 informats 44-45

 length 318-319

 sorting 106-107

 truncation error 318-319

character-values-converted note 310-311

charts, bar 230-231

chi-square statistic with FREQ procedure
264-266

CHISQ option in FREQ procedure 264-266

CHTML destination 292

CI= option in TTEST procedure 260

CLASS statement

 ANOVA procedure 276-279

 MEANS procedure 118

 STYLE= option in TABULATE procedure
170

 TABULATE procedure 124-133

 TTEST procedure 260

CLASSLEV statement in TABULATE procedure
172

CLM option in fitted curves 240-241

CLM option in MEANS procedure 258-259

CLMTRANSPARENCY= option in fitted curves
240

CLOSE option

 ODS HTML statement 158-159

 ODS LISTING statement 156-157

 ODS PDF statement 162-163

 ODS RTF statement 160-161

Cochran-Armitage test 264

Cochran-Mantel-Haenszel statistics 264

Cochran's Q test 264

coded data, custom formats 114-115

coefficient of variation

 ANOVA procedure 278-279

 MEANS procedure 258

 REG procedure 272

COLAXIS statement 248

collating sequence 106

colon informat modifier 48-49

color

 graph attributes 246

 PRINT procedure 166-167

 REPORT procedure 168-169

 style attributes 174-175

 style templates 151

 TABULATE procedure 170-171

COLOR= option for graph attributes 246

- COLOR= option in TITLE statement 164
- Column Attributes window 32
- COLUMN location in STYLE= option 168-169
- column pointers
 - @' character' 48-49
 - @n 46-47, 290-291
 - +n 43
- COLUMN statement in REPORT procedure 134-143
- column-style input 40-41
- columns of data
 - definition 4
 - Viewtable window 32-33
- COLUMNS= option
 - ODS PDF statement 162
 - ODS RTF statement 160-161
 - PANELBY statement 248
- combining SAS data sets
 - concatenating data sets 180-181
 - grand total with original data 190-191
 - interleaving data sets 182-183
 - merging summary statistics 188-189
 - one observation with many 190-191
 - one-to-many match merge 186-189
 - one-to-one match merge 184-185
 - selecting observations during a merge 200-201
 - stacking data sets 180-181
 - updating a master data set 192-193
- command bar in SAS windowing environment 13
- commas
 - reading comma-delimited data 58-61
 - reading numbers containing commas 42, 44-45
 - writing comma-delimited data 284-287, 292-293
 - writing numbers containing commas 112-113
- COMMAw.d format 112-113
- COMMAw.d informat 44-45
- COMMAXw. informat 44-45
- comments
 - *; 3
 - /* */ 3
 - /* */ in z/OS 320
 - unmatched 320-321
- comparison operators 82-83, 102-103
- compile and execute phases 222
- COMPRESS function 78-79
- COMPRESS= data set option 323
- COMPUTE statement in REPORT procedure 144-145
- COMPUTED usage option 136, 144-145
- concatenating SAS data sets 180-181
- concatenation
 - function 78-79
 - operator, || 79
- conditional statements
 - macro 220-221
 - standard 82-87
- confidence limits 258-264
 - plotting 240-241
- CONFIDENCE option in CORR procedure 270
- constants
 - ASCII 58
 - character 74
 - date 88-89
 - hexadecimal 58
 - name 5
 - numeric 74
- CONTAINS operator 102
- CONTENTS procedure 70-71
 - debugging programs 317
 - POSITION option 96
- Contents window 22
- CONTENTS= option in ODS HTML statement 158-159
- converting character to numeric and vice versa 310-311
- COOKSD option in REG procedure 274
- CORR procedure 268-271
- correlations 268-269
- counts, frequency 122-123, 126-127, 142-143, 146-147

CREATE statement in SQL procedure 326
 cross-tabulations 122-123, 126-127, 146-147
 CSS option in MEANS procedure 258
 CSV destination 292-293
 CSV files
 reading 58-61
 writing 284-287, 292-293
 CSV value in the DBMS= option
 EXPORT procedure 286
 IMPORT procedure 60
 CSVALL destination 292
 CUMFREQPLOT option in FREQ procedure 266
 cumulative distribution function plots 256
 cumulative totals
 FREQ procedure 123
 sum statement in DATA step 92-93
 CURVELABEL= option
 fitted curves 240
 SERIES statement 238
 custom formats, FORMAT procedure 114-115
 CV option in MEANS procedure 258

D

DATA _NULL_
 writing custom reports 116-117
 writing raw data files 290-291
 data dictionary 70-71
 data engines 31
 data entry with Viewtable window 32-33
 DATA location in STYLE= option 166-167
 data set options
 compared to statement options 198-199
 compared to system options 198-199
 COMPRESS= 323
 DROP= 198-199, 323
 FIRSTOBS= 198-199, 297
 IN= 198-201
 KEEP= 198-199, 323
 LABEL= 70-71
 OBS= 198-199, 297
 RENAME= 198-199
 data sets, SAS
 changing observations to variables 204-205

combining a grand total with data 190-191
 combining one observation with many
 190-191
 compressing 323
 concatenating 180-181
 contents of 22-25, 70-71
 creating from procedure output 154-155
 definition 4
 interleaving data sets 182-183
 inverting, TRANSPOSE procedure 204-205
 LABEL= data set option 198-199
 merging summary statistics 188-191
 merging, one-to-many 186-189
 merging, one-to-one 184-185
 modifying a single data set 178-179
 names 5
 options 70-71, 198-203, 329-330
 permanent 64-69
 permanent, examples 103, 178-179, 192-193,
 289
 printing 108-109
 properties of 24-25
 reading a single data set 178-179
 saving 64-69
 saving summary statistics to 120-121,
 154-155
 selecting observations during a merge
 200-201
 size 5
 sorting 104-107
 stacking data sets 180-181
 subsetting IF statement 86-87
 subsetting OUTPUT statement 194-195
 subsetting WHERE statement 102-103,
 328-330
 subsetting WHERE= data set option 198,
 202-203, 329-330
 temporary versus permanent 64-65
 updating a master data set 192-193
 Viewtable window 22-23, 32-33
 WORK library 20, 64-65
 writing multiple data sets 194-195

- DATA statement 6-7
 - _NULL_ data set name 116-117, 290-291
 - multiple data sets 194-195
 - permanent data sets 64-69
- DATA step 6-9
 - built-in loop 8-9
 - combining SAS data sets 180-201
 - creating and modifying variables 74-97
 - definition 6
 - reading a single SAS data set 178-179
 - reading raw data files 30-31, 36-59
 - writing raw data files 290-291
 - wrong results, no message 312-313
- data types 4
 - assignment statements 74-75
 - converting, character to numeric 310-311
 - converting, numeric to character 310-311
- data, reading 30-31, 34-63
 - column style 40-41
 - comma-separated values 58-61
 - delimited data 58-61
 - internal 36
 - messy data 48-49
 - methods for getting into SAS 30-31
 - missing data at end of line 57
 - mixing input styles 46-47
 - multiple lines of data per observation 50-51
 - multiple observations per line of data 52-53, 197
 - non-standard format 42-43
 - part of a data file 54-55, 297
 - PC files 34-35, 62-63
 - skipping lines of raw data 50-51, 56
 - skipping over variables 40-41
 - space-delimited 38-39
 - variable length records 57
 - variable length values 48-49
- data, writing 282-293
 - delimited 284-287, 290-293
 - methods 282-283
 - PC files 284, 288-289
 - raw data 284-287, 290-293
- DATA= option 100, 198-199
- DATAFILE= option in IMPORT procedure 60-63
- DATALABEL= option
 - SCATTER statement 236
 - SERIES statement 238
 - VBAR or HBAR statement 230
- DATALINES statement 36
- DATAROWS= statement in the IMPORT procedure 60
- DATASTMTCHK= system option 301
- DATE system option 27
- DATEJUL function 80-81, 90-91
- dates 88-91
 - automatic macro variables 214-215
 - constants 88-89
 - converting dates 80-81, 88-89
 - definition of a SAS date 88
 - formats, table of 90-91, 112-113
 - functions, table of 80-81, 90-91
 - informats, table of 44-45, 90-91
 - Julian dates 90-91
 - printing current date on output 27
 - reading raw data with 42-43
 - setting default century 88
 - today's date 80-81, 88-91
- DATETIMEw. informat 44-45
- DATETIMEw.d format 112-113
- DATEw. format 90-91, 112-113
- DATEw. informat 44-45, 90-91
- DAY function 80-81, 90-91
- dBase files
 - reading 34-35
 - writing 284
- DBMS= option
 - EXPORT procedure 286-289
 - IMPORT procedure 60-63
- DDMMYYw. informat 44-45, 90-91
- debugging SAS programs 296-323
 - avoiding errors 296-297
 - fixing errors 298-299
 - INPUT reached past end of line 302-303, 305
 - invalid data 305-307
 - invalid option 314-315

debugging SAS programs (*continued*)

- lost card 304-305
- macros 224-225
- missing semicolon 300-301
- missing values were generated 308-309
- option not recognized 314-315
- out of memory or disk space 322-323
- SAS stops in middle of job 320-321
- statement not valid 314-315
- truncation of character data 318-319
- values have been converted 310-311
- variable not found 316-317
- variable uninitialized 316-317
- wrong results, no message 312-313

decimal places

- printing data 110-111
- reading data 42-43

DEFINE statement in REPORT procedure

136-139, 144-145

DELETE statement 86-87

deleting

- observations 86-87
- variables 198-199

delimited data

- reading 58-61
- writing 284-287, 292-293

DELIMITER= option

- FILE statements 290
- INFILE statements 58-59

DELIMITER= statement in IMPORT procedure

60-61

density curves 232-233

DENSITY statement in SGPLOT procedure

232-233

DESCENDING option in SORT procedure

104-105

descriptive statistics 118-143, 254-259

destinations, output

- CSV 150
- DOCUMENT 150
- for graphics 229, 250-251
- HTML 150-151, 158-159
- LISTING 150, 156-157, 229

MARKUP 150

OUTPUT 150

PDF 150, 162-163

PRINTER 150-151

PS 150

RTF 150-151, 160-161

XML 150

DEVIATIONPLOT option in FREQ procedure

266

DIAGNOSTICS option in REG procedure

274-275

dictionary, data 70-71

dimensions in TABULATE procedure 124-127

dimensions of graph images 250

DISCRETE option in XAXIS or YAXIS statement

242-243

DISCRETEOFFSET= option in VBAR or HBAR

statement 230

disk space, running out of 322-323

Display Manager 10-23

DISPLAY usage option 136

dividing

data file 54-55

SAS data set 194-195, 202-203, 329-330

DLM value in the DBMS= option

EXPORT procedure 286

IMPORT procedure 60

DLM= option

FILE statements 290

INFILE statements 58-59

DLMSTR= option in INFILE statements 58

DO statement 82-83

arrays 94-95

with OUTPUT statement 196

DOCUMENT destination 150

documenting

data sets 70-71, 198-199

programs 3

dollar signs

printing data 112-113

reading data 42, 44-45

DOLLARw.d format 112-113

DOWN= option in KEYLEGEND statement 244

DPI= option in ODS statement 251
 DROP= data set option 198-199, 323
 DSD option
 FILE statements 290
 INFILE statements 58-59
 DTDATE*w*. format 112-113
 DTRESET option in ODS RTF statement 161
 DUNCAN option in ANOVA procedure 276
 Duncan's multiple range test 276
 duplicate observations, eliminating 104-105
 DUPOUT= option in SORT procedure 104

E

EBCDIC sort order 106
 editing data with Viewtable window 32-33
 editing graphs 229
 editor 12, 14-15
 RECALL command 15
 SUBMIT command 14
 Syntax Sensitive 297
 ELLIPSE= option in CORR procedure 270
 ELSE statement 84-85
 END statement 82-83
 ENDCOMP statement in REPORT procedure 144-145
 engines, data 31
 Enhanced Editor 12, 14-15
 entering data with Viewtable window 32-33
 EQ comparison operator 82, 102
 equations
 assignment statements 74-75
 generating data 196
 errors
 avoiding errors 296-297
 fixing errors 298-299
 INPUT reached past end of line 302-303, 305
 invalid data 306-307
 invalid option 314-315
 lost card 304-305
 missing semicolon 300-301

missing values were generated 308-309
 option not recognized 314-315
 out of memory or disk space 322-323
 SAS stops in middle of job 320-321
 statement not valid 314-315
 truncation of character data 318-319
 values have been converted 310-311
 variable not found 316-317
 variable uninitialized 316-317
 wrong results, no message 312-313
 EURDFDD*w*. format 90-91, 112-113
 EUROX*w.d* format 112-113
 E*w*. format 112-113
 EXACT option in FREQ procedure 264
 EXCEL engine 31
 Excel files
 reading 34-35, 62-63
 writing 284, 288-289
 excluding output objects 153
 executing SAS programs
 methods 10-11
 SAS windowing environment 14-15
 Explorer window 12-13, 20-22
 EXPONENTIAL option in distribution plots 256
 EXPORT procedure
 delimited files 286-287
 Excel files 288-289
 WHERE= data set option 202-203
 Export Wizard 284-285
 exporting data 282-293
 delimited files 284-287, 290-293
 methods 282-283
 PC files 284, 288-289
 raw data files 284-287, 290-293
 to other software 284, 288
 expressions
 mathematical 74-75
 using dates 88-89
 using functions 76-77
 external data 36-37
 EXTREME option in HBOX or VBOX statement 234

F

F value

- ANOVA procedure 278-279
- REG procedure 272

FILE statement

- DLM= option 290
- DSD option 290
- PRINT option 116-117
- writing raw data files 290-291
- writing reports 116-117

FILE= option

- for graphics 250
- ODS HTML statement 158-159
- ODS LISTING statement 156-157
- ODS PDF statement 162-163, 250
- ODS RTF statement 160-161, 250

FILLATTRS= option for graph attributes 246

FIRST.byvariable 206-207

FIRSTOBS= option

- data set option 198-199, 297
- INFILE statement 56, 297

Fisher's exact test 264

fit plots 274-275

FITPLOT option in REG procedure 274-275

fitted curves 240-241

flat files 30, 36

FLYOVER style attribute 174-175

font

- graph attributes 246-247
- style attributes 174-175

FONT_FACE style attribute 174-175

FONT_SIZE style attribute 174-175

FONT_STYLE style attribute 174-175

FONT_WEIGHT style attribute 174-175

FONT= option in TITLE statement 164-165

FOOTNOTE statement 100-101, 164-165

FOREGROUND style attribute 174-175

FORMAT procedure 114-115

- grouping with 146-147
- with SGPLOT procedure 231, 267
- with TABULATE procedure 130-131

FORMAT statement 70-71, 110-111

- DATA step compared to PROC step 110

FORMAT= option in TABULATE procedure
128-129, 132-133

formats

- ATTRIB statement 318
- dates 89-91
- FORMAT statement 70-71, 110-111
- grouping with 146-147
- input formats 42-45
- table of 112-113
- use 110-111
- user-defined 114-115, 146-147

formatted style input 42-43

FRAME= option in ODS HTML statement
158-159

free formatted style input 38-39

FREQ procedure 122-123, 264-267

FREQPLOT option in FREQ procedure 266-267

frequency tables 122-127, 142-143, 264-265

functions

- dates 88-91
- INPUT function 311
- PUT function 311
- table of 78-81, 90-91
- use 76-77

G

gamma 264

GAMMA option in distribution plots 256

GE comparison operator 82, 102

generating data

- DO and OUTPUT statements 196

GETNAMES= statement 60-61

GIF image format 250

global macro variables 210-211

GPATH= option in ODS statement 251

GRANDTOTAL location in STYLE= option 166

graphics, ODS 228-229

- ANOVA procedure 276-277

- CORR procedure 270-271

- FREQ procedure 266-267

- image formats 250

- image properties 250-251

- insets 244-245

- legends 244-245
- REG procedure 274-275
- saving graphs 250-251
- SGPANEL procedure 248-249
- SGPLOT procedure 230-241
- style attributes 246-247
- TTEST procedure 262-263
- UNIVARIATE procedure 256-257
- GRID option in XAXIS or YAXIS statement 242
- GROUP usage option 136, 138-139
- GROUP= option
 - fitted curves 240
 - HBOX or VBOX statement 234
 - SCATTER statement 236-237
 - SERIES statement 238
 - VBAR statement 230
- GROUPDISPLAY= option in VBAR or HBAR
 - statement 230-231
- GROUPTHORIZONTAL option in FREQ
 - procedure 266-267
- grouping observations
 - BY statement 100
 - FREQ procedure 122-123
 - IF-THEN/ELSE statements 84-85
 - MEANS procedure 118-119
 - REPORT procedure 136-139
 - SORT procedure 104-107
 - TABULATE procedure 124-125
 - traffic-lighting 172-173
 - user-defined format 146-147, 172-173
- GT comparison operator 82, 102
- GUESSINGROWS= statement in the IMPORT
 - procedure 61

H

- H0= option in TTEST procedure 260
- HBAR statement in SGPLOT procedure 230
- HBOX statement 234
- HEADER location in STYLE= option 166-168
- headers
 - changing in TABULATE output 130-131
 - reading raw data 56
 - specifying style for 166-171

- HEIGHT= option
 - ODS GRAPHICS statement 250-251
 - TITLE statement 164-165
- hexadecimal data
 - constants 58
- HIGH keyword in FORMAT procedure 114-115
- HISTOGRAM option
 - CORR procedure 270
 - TTEST procedure 262
- HISTOGRAM statement
 - SGPLOT procedure 232-233
 - UNIVARIATE procedure 256-257
- histograms 232-233, 256-257, 262, 270
- HOEFFDING option in CORR procedure 268
- HTML data files, writing 292-293
- HTML output 150-151, 158-159
- hypertext links, style attribute 174-175
- HyperText Markup Language 150-151, 158-159

I

- ID statement
 - BY statement with ID 189
 - PRINT procedure 108
 - TRANSPPOSE procedure 204-205
- IF statement, subsetting 86-87
- IF-THEN statements 82-83
- IF-THEN/ELSE statements 84-85
- IMAGE_DPI= option in ODS statement 251
- IMAGENAME= option in ODS GRAPHICS
 - statement 250-251
- images
 - inserting in output 174-175
 - saving 250-251
- IMPORT procedure
 - delimited files 60-61
 - PC files 62-63
 - WHERE= data set option 202-203
- Import Wizard 34-35
- importing data
 - delimited 34-35, 60-61
 - from other software 34-35, 62-63
 - methods 30-31
 - PC files 34-35, 62-63

- IN operator 102
 - in-stream data 36
 - IN= data set option 198-201
 - indentation in SAS programs 3
 - INDEX function 78-79
 - INFILE statement 36-37
 - DELIMITER= option 58-59
 - DLM= option 58-59
 - DSD option 58-59
 - examples by operating environment 36-37
 - FIRSTOBS= option 56, 297
 - LRECL= option 37
 - MISSOVER option 57, 303
 - OBS= option 56, 297
 - TRUNCOVER option 57, 303
 - INFORMAT statement 70-71
 - informats
 - ATTRIB statement 318
 - colon modifier 48-49
 - dates 88-91
 - INFORMAT statement 70-71
 - invalid data 306-307
 - table of 44-45
 - use 42-43
 - input formats 42-47
 - INPUT function 310-311
 - INPUT reached past end of line
 - message in log 37, 302-303, 305
 - INPUT statement
 - column style 40-41
 - data with embedded blanks 40-41
 - delimited data 58-59
 - formatted style 42-45
 - free formatted 38-39
 - list style 38-39
 - mixing input styles 46-47
 - multiple INPUT statements 54-55, 197
 - multiple lines per observation 50-51
 - multiple observations per line 52-53
 - reading blanks as missing 40-41
 - reading non-standard data 42-43
 - reading part of a raw data file 54-55
 - skipping lines of raw data 50-51
 - skipping over variables 40-41
 - space-delimited 38-39
 - INSERT statement in SQL procedure 326-328
 - INSET statement 244-245
 - INT function 80-81
 - integer data
 - data types 4
 - truncating decimal places 80-81
 - interleaving SAS data sets 182-183
 - internal data 36
 - internet browser, creating files for 158-159
 - INTERVALPLOT option in TTEST procedure 262
 - invalid data message in log 306-307
 - lost card note 305
 - invalid option message in log 314-315
 - inverting data sets 204-205
 - IS NOT MISSING operator 102
 - ITALIC option in TITLE statement 164-165
 - italics, explanation of usage xiii
 - iterative logic 94-95
- ## J
- JMP files
 - reading 34
 - writing 284
 - JPEG image format 250
 - Julian dates 90-91
 - JULIAN*w.* format 90-91, 112-113
 - JULIAN*w.* informat 44-45, 90-91
 - JUST style attribute 169, 171, 174-175
 - justification
 - character variables 78-79
 - output 27
 - style attributes 169, 171, 174-175
 - titles and footnotes 164-165
 - JUSTIFY= option in TITLE statement 164-165
- ## K
- kappa statistics 264
 - KAPPAPLOT option in FREQ procedure 266
 - KEEP= data set option 198-199, 323
 - KENDALL option in CORR procedure 268

- Kendall's tau-b 264, 268
- kernel density plot 232-233
- KERNEL option in DENSITY statement 232-233
- KEYLEGEND statement 244-245
- kurtosis
 - MEANS procedure 258
 - UNIVARIATE procedure 254-255
- KURTOSIS option in MEANS procedure 258
- L**
- LABEL option in PRINT procedure 108
- LABEL statement 70-71, 101
 - SGPLOT procedure 231
 - TABULATE procedure 130
- LABEL= option
 - data set option 198-199
 - REFLINE statement 242-243
 - XAXIS or YAXIS statement 242-243
- LABELATTRS= option for graph attributes 246-247
- labels
 - ATTRIB statement 318
 - data set 70-71, 198-199
 - value 114-115
 - variable 70-71, 101
 - variable in Viewtable 23
- lambda 264
- LAST.byvariable 206-207
- LCLM option in MEANS procedure 258
- LEFT function 78-79
- LEFTMARGIN= system option 27
- legends for graphs 244-245
- length of a variable 71, 318-319, 322-323
- LENGTH statement
 - character data 318-319, 322-323
 - numeric data 322-323
- LENGTH= option in REPORT procedure 144-145
- LIBNAME statement 66-67
- library, SAS data 20-22, 64-69
- libref 21, 64-69
- LIMITSTAT= option in VBAR or HBAR statement 230
- line plots 238-239
- line pointers
 - / 50-51, 290
 - #n 50-51, 290
- line-hold specifiers
 - @ compared to @@ 55
 - @, trailing 54-55, 290
 - @@, double trailing 52-53
- line, graph attributes 246-247
- LINEAR option in XAXIS or YAXIS statement 242
- LINEATTRS= option for graph attributes 246-247
- LINGUISTIC sort option 106-107
- links, style attributes for hypertext 174-175
- LIST option in FREQ procedure 122
- list style input 38-39
- LISTING output 150, 156-157, 229, 250
- lists, variable names 96-97
- local macro variables 210-211
- LOCATION= option in KEYLEGEND statement 244-245
- locations in STYLE= option 166-169
- loess curves 240-241
- LOESS statement in SGPLOT procedure 240-241
- LOG function 80-81
- LOG option in XAXIS or YAXIS statement 242
- Log window
 - SAS windowing environment 12-13, 14-15
- log, SAS 16-17
 - errors, warnings, and notes 298-299
 - notes when reading raw data 37
 - notes when writing raw data files 291
 - writing in with PUT statements 312-313
- LOG10 function 80-81
- logarithmic functions 80-81
- logical operators 82-83, 102-103
- logical record length of raw data files 37
- LOGNORMAL option in distribution plots 256
- loop
 - DATA step, built-in 8-9
 - DO loop 94-95
- lost card note in log 304-305

LOW keyword in FORMAT procedure 114
 LRECL= option in INFILE statements 37
 LT comparison operator 82, 102

M

MACRO system option 211
 macros 210-225
 &SYSDATE macro variable 214-215
 &SYSDAY macro variable 214
 &SYSNOBS macro variable 214
 %DO statements 220-221
 %ELSE statement 220-221
 %END statement 220-221
 %IF-%THEN statements 220-221
 %LET statement 212-215
 %MACRO statement 216-219
 %MEND statement 216-217
 %THEN statement 220-221
 autocall libraries 217
 automatic macro variables 214-215
 CALL SYMPUT 222-223
 concepts 210-211
 debugging errors 224-225
 invoking 216
 local versus global variables 210-211
 MACRO system option 211
 macro variables, definition 210
 MERROR system option 224-225
 MLOGIC system option 224-225
 MPRINT system option 224-225
 parameters 218-219
 quotation marks 224
 SAS macro processor 210-211
 SERROR system option 224-225
 SYMBOLGEN system option 224-225
 MARKERATTRS= option for graph attributes 246-247
 MARKERS option in SERIES statement 238-239
 MARKUP destination 150
 master data set definition 192
 match merging
 IN= data set option 198-201
 one-to-many match merge 186-189

 one-to-one match merge 184-185
 summary statistics 188-191
 mathematical expressions 74-75
 MATRIX option in CORR procedure 270-271
 MAX function 80-81
 MAX keyword
 REPORT procedure 142
 TABULATE procedure 126
 MAX option in MEANS procedure 118
 MAXDEC= option in MEANS procedure 118-119
 maximum value
 across observations 80-81
 across variables 92-93, 118-119, 206-207
 FIRST. and LAST. byvariable 206-207
 HBOX or VBOX statement 234-235
 MAX function 80-81
 MEANS procedure 118-119
 REPORT procedure 142
 RETAIN statement 92-93
 TABULATE procedure 126
 UNIVARIATE procedure 255
 McNemar's test 264
 MDY function 80-81, 90-91
 MEAN function 80-81
 missing data 309
 MEAN keyword
 REPORT procedure 142-143
 TABULATE procedure 126-127
 MEAN option in MEANS procedure 118
 mean square
 ANOVA procedure 278-279
 REG procedure 272
 means
 HBOX or VBOX statement 234-235
 MEAN function 80-81
 MEANS procedure 118-119
 multiple comparisons 276-279
 pairwise comparisons 260-263
 REPORT procedure 142-143
 TABULATE procedure 126-127
 testing 260-263
 UNIVARIATE procedure 255

- MEANS procedure 118-121, 188-191, 258-259
- MEANS statement, ANOVA procedure 276-279
- MEASURES option in FREQ procedure 264, 266
- median
 - HBOX or VBOX statement 234-235
 - MEANS procedure 118, 258-259
 - REPORT procedure 142
 - TABULATE procedure 126
 - UNIVARIATE procedure 254-255
- MEDIAN option in MEANS procedure 118, 258-259
- memory, running out 322-323
- menus, pull-down and pop-up 13
- MERGE statement 184-189
 - BY statement 184-189
 - IN= data set option 198-201
 - one-to-many match merge 186-189
 - one-to-one match merge 184-185
 - summary statistics 188-189
- MERROR system option 224-225
- messy raw data, reading 48-49
- Microsoft Excel files
 - reading 34, 62-63
 - writing 284, 288-289
- MIN function 80-81
- MIN keyword
 - REPORT procedure 142
 - TABULATE procedure 126
- MIN option in MEANS procedure 118
- minimum value
 - across observations 80-81
 - across variables 92-93, 118-119, 206-207
 - FIRST. and LAST. byvariable 206-207
 - HBOX or VBOX statement 234-235
 - MEANS procedure 118-119
 - MIN function 80-81
 - REPORT procedure 142
 - RETAIN statement 92-93
 - TABULATE procedure 126
 - UNIVARIATE procedure 255
- missing data values 5
 - assignment statements 75, 308-309
 - end of raw data line 57
 - finding number 118-119, 122-123
- IF-THEN statements 84-85
- match merge 185
- MEANS procedure 118
- reading blanks as 40-41
- REPORT procedure 136-137
- SET statement 181
- SORT procedure 104-105
- TABULATE statement 124
- UPDATE statement 192-193
- MISSING option
 - FREQ procedure 122
 - HBOX or VBOX statement 234
 - MEANS procedure 118
 - PANELBY statement 248
 - REPORT procedure 136-137
 - TABULATE procedure 124
 - VAR or HBAR statement 230
- missing semicolon 300-301
- missing values generated note 75, 308-309
- MISSOVER option in INFILE statements 57, 303
- MISSPRINT option in FREQ procedure 122
- MISSTEXT= option in TABULATE procedure 128-129
- MIXED= statement in the IMPORT procedure 62
- MLOGIC system option 224-225
- MMDDYYw. format 90-91, 112-113
- MMDDYYw. informat 44-45, 90-91
- mode of a variable
 - MEANS procedure 118, 258
 - REPORT procedure 142
 - TABULATE procedure 126
 - UNIVARIATE procedure 254-255
- MODEL statement
 - ANOVA procedure 276-278
 - REG procedure 272-273
- modes of running SAS 10-11
- modifying SAS data sets
 - MERGE statement 184-189
 - SET statement 178-182, 190-191
 - UPDATE statement 192-193
- MONTH function 80-81, 90-91
- MPRINT system option 224-225

multiple comparisons 276-279
 multiple lines per observation, reading 50-51
 multiple observations per line, reading 52-53

N

N function 80-81
 N keyword
 REPORT procedure 142-143
 TABULATE procedure 126
 N option in MEANS procedure 118, 258-259
 NAME= option in graphics statements 244
 names for
 data sets 5, 64
 formats 114
 librefs 21, 64-69
 macro variables 212
 macros 216
 variable lists 96-97
 variables 5
 NBINS= option in HISTOGRAM statement 232
 NE comparison operator 82, 102
 New Library window 65
 NMISS function 80-81
 NMISS keyword
 REPORT procedure 142
 TABULATE procedure 126
 NMISS option in MEANS procedure 118
 NOBORDER option in KEYLEGEND statement 244
 NOBYVAR option in TTEST procedure 260
 NOCENTER system option 27
 NOCOL option in FREQ procedure 122
 NODATE system option 27
 NODUPKEY option in SORT procedure 104-105
 NOLEGCLI option in fitted curves 240
 NOLEGCLM option in fitted curves 240-241
 NOLEGFIT option in fitted curves 240
 NOMARKERS option in fitted curves 240-241
 NOMISSINGGROUP option
 SCATTER statement 236
 SERIES statement 238
 NONE option
 CORR procedure 270

 TTEST procedure 262
 NONUMBER system option 27
 NOOBS option in PRINT procedure 108
 NOPERCENT option in FREQ procedure 122
 NOPRINT option
 MEANS procedure 120-121
 REPORT procedure 145
 SQL procedure 326
 NOPROCTITLE, ODS statement 156-163
 normal density plot 232-233
 NORMAL option
 CDFPLOT statement 256
 DENSITY statement 232-233
 HISTOGRAM statement 256-257
 UNIVARIATE procedure 254
 normality test 254
 NOROW option in FREQ procedure 122
 notes in SAS log 16-17, 298-299
 INPUT reached past end line 37, 302-303, 305
 invalid data 306-307
 lost card 304-305
 missing values were generated 75, 308-309
 values have been converted 310-311
 variable uninitialized 316-317
 NOVARNAM option in PANELBY statement 248-249
 NOWINDOWS option in REPORT procedure 134-135
 NUMBER system option 27
 numbering observations, _N_ variable 206-207
 numeric data
 commas, reading 42, 44-45
 commas, writing 112-113
 converting to character 310-311
 definition 4
 formats 112-113
 functions 80-81
 informat 44-45
 length 322-323
 reading non-standard 42-43
 reading standard 38-41
 numeric values converted note 310-311

NUMERIC_COLLATION= suboption in SORT
procedure 106-107

O

OBS location in STYLE= option 166

OBS= option

data set option 198-199, 297

INFILE statements 56, 297

observations

changing to variables 204-205

combining single observation with many
190-191

creating a numbering variable 206-207

definition 4

deleting 86-87

duplicate, eliminating 104-105

grouping in procedures 146-147

grouping with IF-THEN/ELSE 84-85

interleaving 182-183

making several from one 196-197

merging 184-187

printing 108-109

reading multiple lines per observation
50-51

reading multiple observations per line
52-53

sorting 104-107

subsetting DELETE statements 86-87

subsetting FIRSTOBS= option 198-199

subsetting IF statement 86-87

subsetting IN= data set option 198-201

subsetting OBS= option 198-199

subsetting OUTPUT statements 194-195

subsetting WHERE statements 102-103,
328-330

subsetting WHERE= data set option 198,
202-203, 329-330

tracking with IN= data set option 198-201

updating 192-193

OBSERVEDBYPREDICTED option in REG
procedure 274

OBSHEADER location in STYLE= option 166

odds ratios 264-265

ODDSRATIOPLOT option in FREQ procedure
266

ODS 150-175, 228-257, 266-267, 270-277

ODS CSV statement 292-293

ODS EXCLUDE statement 153

ODS Graphics 228-229

ANOVA procedure 276-277

CORR procedure 270-271

FREQ procedure 266-267

image properties 250-251

insets 244-245

legends 244-245

REG procedure 274-275

saving graphs 250-251

SGPANEL procedure 248-249

SGPLOT procedure 230-241

style attributes 246-247

TTEST procedure 262-263

UNIVARIATE procedure 256-257

ODS Graphics Editor 229

ODS GRAPHICS statement 228, 250-251

ODS HTML statement 158-159, 250, 292-293

ODS LISTING statement 229, 250

ODS NOPROCTITLE statement 158-163

ODS OUTPUT statement 154-155

ODS PDF statement 162-163, 250

ODS RTF statement 160-161, 250

ODS SELECT statement 153

ODS TRACE statement 152-153

one-to-many match merge 186-189

one-to-one match merge 184-185

one-way frequency table 122-123

opening a table in Viewtable window 33

operators

arithmetic 74-75

comparison 82-83, 102-103

logical 82-83, 102-103

option not recognized error in log 314-315

options

comparison of types of options 198-199

data set 70-71, 198-203, 329-330

system 26-27

OPTIONS procedure 26
 OPTION= option 211
 OPTIONS statement 26-27
 macro debugging options 224-225
 Options window in SAS windowing
 environment 27
 OR operator 82-83, 102
 ORDER usage option 136-137
 ordering observations 104-107
 ORIENTATION= system option 27
 OTHER keyword FORMAT procedure 114
 out of disk space message 322-323
 out of memory message 322-323
 out of time, job runs 321
 OUT= option
 FREQ procedure 122
 IMPORT procedure 60-63
 MEANS procedure 120-121
 SORT procedure 104-105
 OUTFILE= option in EXPORT procedure
 286-289
 outliers 234-235, 254-255
 output
 centering 27
 changing appearance of data in 89-91,
 110-115
 creating SAS data sets from 154-155
 customizing with STYLE= option 150-175
 footnotes 100-101, 164-165
 graphics 228-229
 HTML 158-159, 229
 labels 101
 LISTING 156-157, 229
 location for 15, 18-19
 PDF 162-163, 229
 RTF 160-161, 229
 text 156-157
 titles 100-101, 164-165
 titles, removing 100-101, 156-163
 Output Delivery System 150-175, 228-257,
 266-267, 270-277
 OUTPUT destination 150, 154-155
 output object 151-155

OUTPUT statement
 DATA step 194-197
 DO statement 196
 MEANS procedure 120-121
 multiple observations from one 196-197
 writing multiple data sets 194-195
 Output window 12-13, 15, 156-157
 OUTPUTFMT= option in ODS GRAPHICS
 statement 250-251

P

P1 option in MEANS procedure 258
 P10 option in MEANS procedure 258
 P25 option in MEANS procedure 258
 P5 option in MEANS procedure 258
 P50 option in MEANS procedure 258
 P75 option in MEANS procedure 258
 P90 option in MEANS procedure 258
 P95 option in MEANS procedure 258
 P99 option in MEANS procedure 258
 page breaks
 ODS output 160-163
 PUT statement 116-117
 PAGE option in REPORT procedure 140
 PAGE= option in ODS HTML statement 158-159
 PAGENO= system option 27
 PAIRED statement in TTEST procedure 260-261
 pairwise t test 260-263
 PANELBY statement 248-249
 parameter estimates 273
 parameters, macro 218-219
 PATTERN= option for graph attributes 246
 PBSPLINE statement in SGPLOT procedure 240
 PC files
 reading 34, 62
 writing 284, 288-289
 PC Files Server 288
 PCTN keyword
 REPORT procedure 142
 TABULATE procedure 126
 PCTSUM keyword
 REPORT procedure 142
 TABULATE procedure 126

- PDF image format 250
- PDF output 150, 162-163
- Pearson coefficient 264, 268-269
- percentages
 - calculating in DATA step 188-189
 - FREQ procedure 122-123
 - REPORT procedure 142
 - TABULATE procedure 126
- percentiles
 - HBOX or VBOX statement 234-235
 - MEANS procedure 258
 - REPORT procedure 142
 - UNIVARIATE procedure 255
- PERCENT*w*. informat 44-45
- PERCENT*w.d* format 112-113
- permanent SAS data sets 64-69
 - examples 103, 178-179, 192-193, 289
- plots 228-257, 266-267, 270-277
- PLOTS= option
 - CORR procedure 270-271
 - FREQ procedure 266-267
 - REG procedure 274-275
 - TTEST procedure 262-263
- PNG image format 250
- pointers
 - @'character' column pointer 48-49
 - @n column pointer 46-47, 290-291
 - / line pointer 50-51, 116-117, 290
 - #n line pointer 50-51, 290
 - +n column pointer 42-43
 - INPUT statements 42-43, 46-49, 50-51
 - PUT statements 116-117, 290-291
- POSITION option in CONTENTS procedure 96
- POSITION= option in INSET or KEYLEGEND
 - statement 244-245
- POSTIMAGE style attribute 174-175
- PostScript output 150, 250
- POSTTEXT style attribute 174-177
- PLOT statement in UNIVARIATE procedure 256
- precedence, mathematical rules 74
- predicted values in regression 272-273
- Preferences window 156-157
- PREIMAGE style attribute 174-175
- PRETEXT style attribute 174-175
- PRIMARY option in SORT procedure 106-107
- print formats 110-115
 - user-defined 114-115
- PRINT option in FILE statements 116-117
- PRINT procedure 108-109
 - BY and ID together 189
 - STYLE= option 166-167
- printed values, changing appearance 110-111
- PRINTER
 - output 150-151
 - style template 151, 162-163
- probability plot 256-257
- probability-probability plot 256
- PROBPLOT statement in UNIVARIATE
 - procedure 256-257
- PROBT option in MEANS procedure 258
- PROC ANOVA 276-279
- PROC CONTENTS 70-71
 - for debugging programs 317
 - POSITION option 96
- PROC CORR 268-271
- PROC EXPORT
 - delimited files 286-287
 - PC files 288-289
 - WHERE= data set option 202-203
- PROC FORMAT 114-115
 - with SGPLOT procedure 231, 267
 - with TABULATE procedure 130-131
- PROC FREQ 122-123, 264-267
- PROC IMPORT
 - delimited files 60-61
 - PC files 62-63
 - WHERE= data set option 202-203
- PROC MEANS 118-121, 188-189, 258-259
- PROC OPTIONS 26
 - OPTION= option 211
- PROC PRINT 108-109
 - BY and ID together 189
 - STYLE= option 166-167
- PROC REG 272-275

PROC REPORT 134-145
 STYLE= option 168-169
 PROC SGPanel 228, 248-249
 PROC SGPLOT 228, 230-247
 PROC SORT 104-107
 PROC SQL 325-328
 PROC statement 6-7, 100
 DATA= option 100
 PROC step
 common statements and options 100-101
 definition 6-7
 PROC SUMMARY 121
 PROC TABULATE 124-133
 CLASSLEV statement 170
 STYLE= option 170-171
 PROC TRANSPOSE 204-205
 PROC TTEST 260-263
 PROC UNIVARIATE 254-257
 procedures
 common statements and options 100-101
 definition 6-7
 title, removing 156-163
 PROFILESLOT option in TTEST procedure 262
 Program Editor 12, 14-15
 RECALL command 15
 SUBMIT command 14
 PROPCASE function 78-79
 Properties window 24-25
 PS destination 150-151
 PS image format 250
 PUT function 310-311
 PUT statement
 ALL variable name list 312-313
 PAGE keyword 116-117
 debugging with 312-313
 formats 110-113
 writing a raw data file 290-291
 writing in SAS log 312-313
 writing reports 116-117
 PUTLOG statement 312-313

Q

Q1 option in MEANS procedure 258

Q3 option in MEANS procedure 258
 QQPLOT option
 REG procedure 274
 TTEST procedure 262-263
 QQPLOT statement in UNIVARIATE procedure 256
 QTR function 80-81, 90-91
 quantile-quantile plot 256, 262-263
 quantiles 234-235, 254-258
 question mark informat modifier, double 307
 QUIT statement 7
 quotation marks
 FOOTNOTE statements 100-101
 in macros 212
 reading delimited data with 58-61
 TITLE statements 100-101
 unmatched 320

R

R-square
 ANOVA procedure 278-279
 REG procedure 272-273
 RANGE option in MEANS procedure 118, 258
 RANGE= statement in the IMPORT procedure 62
 RBREAK statement in REPORT procedure 140-141
 reading data 30-31, 34-63
 column style 40-41
 comma-separated values 58-61
 delimited data 58-61
 internal 36
 messy data 48-49
 methods for getting into SAS 30-31
 missing data at end of line 57
 mixing input styles 46-47
 multiple lines of data per observation 50-51
 multiple observations per line of data 52-53, 197
 non-standard format 42-43
 part of a data file 54-55, 297
 PC files 34, 62
 skipping lines of raw data 50-51, 56

- skipping over variables 40-41
 - space-delimited 38-39
 - variable length records 57
 - variable length values 48-49
- reading SAS data sets
 - a single data set 178-179
 - concatenating data sets 180-181
 - interleaving data sets 182-183
 - merging summary statistics 188-191
 - one-to-many match merge 186-189
 - one-to-one match merge 184-185
 - stacking data sets 180-181
 - updating a master data set 192-193
- RECALL Program Editor command 15
- record length of raw data files 37
- REFLINE statement 242-243
- REG procedure 272-275
- REG statement in SGPLOT procedure 240-241
- regression 272-273
 - lines, plotting 240-241
- relative risk measures 264
- REL RISK option in FREQ procedure 264, 266
- REL RISK PLOT option in FREQ procedure 266
- RENAME= data set option 198-199
- REPLACE option
 - EXPORT procedure 286-289
 - IMPORT procedure 60-63
- REPORT procedure 134-145
 - STYLE= option 168-169
- reports
 - controlling style of 150-151, 166-175
 - PRINT procedure 108-115
 - REPORT procedure 134-145
 - TABULATE procedure 124-133
 - writing custom 116-117
- RESET option in ODS GRAPHICS statement 250-251
- RESIDUAL HISTOGRAM option in REG procedure 274
- RESIDUALS option in REG procedure 274
- RESPONSE= option in VBAR statement 230
- results
 - centering 27
 - changing appearance of data in 89-91, 110-115
 - creating SAS data sets from 154-155
 - customizing with STYLE= option 150-175
 - footnotes 100-101, 164-165
 - graphics 228-229
 - HTML 158-159, 229
 - labels 101
 - LISTING 156-157, 229
 - location for 15, 18-19
 - PDF 162-163, 229
 - RTF 160-161, 229
 - titles 100-101, 164-165
 - titles, removing 100-101, 158-163
- Results Viewer window 12-13, 18-19
- Results window 12-13, 15, 18-19
 - for graphics 229
- RETAIN statement 92-93
- RF PLOT option in REG procedure 274
- RIGHT MARGIN= system option 27
- risk ratios 264
- RISK DIFF PLOT option in FREQ procedure 266
- ROUND function 80-81
- ROW=FLOAT option in TABULATE procedure 130-131
- ROW AXIS statement 248
- rows of data
 - definition 4
 - Viewtable window 32-33
- ROWS= option in PANEL BY statement 248
- RSTUDENT BY LEVERAGE option in REG procedure 274
- RSTUDENT BY PREDICTED option in REG procedure 274
- RTF
 - output 150-151, 160-161
 - style template 151, 160-161
- RULE, invalid data message 306-307
- RUN statement 6-7
 - CALL SYMPUT 222-223
 - missing 321

running SAS programs
 methods 10-11
 SAS windowing environment 14-15

S

SAS automatic variables
 ERROR 206
 N 206-207
 FIRST.byvariable 206-207
 LAST.byvariable 206-207
 macro 214-215
 SAS data library 20-22, 64-69
 SAS data sets
 changing observations to variables 204-205
 combining a grand total with data 190-191
 combining one observation with many
 190-191
 compressing 323
 concatenating 180-181
 contents of 22-25, 70-71
 creating from procedure output 154-155
 definition 4
 interleaving data sets 182-183
 inverting, TRANSPOSE procedure 204-205
 LABEL= data set option 198-199
 merging summary statistics 188-191
 merging, one-to-many 186-189
 merging, one-to-one 184-185
 modifying a single data set 178-179
 names 5
 options 70-71, 198-203, 329-330
 permanent 64-69
 permanent, examples 103, 178-179, 192-193,
 289
 printing 108-109
 properties of 24-25
 reading a single data set 178-179
 saving 64-69
 saving summary statistics to 120-121,
 154-155
 selecting observations during a merge
 200-201
 size 5

 sorting 104-107
 stacking data sets 180-181
 subsetting IF statement 86-87
 subsetting OUTPUT statement 194-195
 subsetting WHERE statement 102-103,
 328-330
 subsetting WHERE= data set option 198,
 202-203, 329-330
 temporary versus permanent 64-65
 updating a master data set 192-193
 Viewtable window 22-23, 32-33
 WORK library 20, 64-65
 writing multiple data sets 194-195
 SAS dates 88-91
 automatic macro variables 214-215
 constants 88-89
 converting dates 80-81, 88-91
 definition of a SAS date 88
 formats, table of 90-91, 112-113
 functions, table of 80-81, 90-91
 informats, table of 44-45, 90-91
 Julian dates 90-91
 printing current date on output 27
 reading raw data with 42-43
 setting default century 88
 today's date 80-81, 88-91
 SAS Enterprise Guide 11, 30
 SAS Explorer 20-22
 SAS functions
 dates 88-91
 INPUT function 311
 PUT function 311
 table 78-81
 use 76-77
 SAS Institute x
 SAS language rules 2-3
 SAS listing 150, 156-157, 229, 250
 SAS log 16-17
 errors, warnings, and notes 298-299
 notes when reading raw data 37
 notes when writing raw data files 291
 writing in log with PUT statements 312-313
 SAS macro processor 210-211

- SAS names, rules for 5
- SAS programs
 - capitalization xiii, 5
 - comments 3
 - compared to SQL 325-330
 - data driven 222-223
 - debugging 296-323
 - definition 2
 - documenting 3
 - finding missing semicolons 300-301
 - fixing 298-299
 - indentation xiii, 3
 - major parts 6-7
 - recalling in Program Editor 15
 - submitting 10-11, 14-15
 - testing 296-297, 299
- SAS windowing environment 10-25
 - command bar 13
 - editor 12, 14-15
 - executing programs from 14-15
 - Options window 27
 - Output window 12-13, 15, 156-157
 - RECALL command 15
 - running programs 14-15
 - SUBMIT command 14-15
- SAS, modes of running 10-11
- SAS/ACCESS 31, 34, 62-63
- SASDATE option in ODS RTF statement 160
- SASHELP library 20-21
- SASUSER library 20-21
 - in the Viewtable window 33
 - saving SAS data sets 64-69
- saving images 250-251
- Scalable Vector Graphics image format 250
- SCALE = option in HISTOGRAM statement 232-233
- SCATTER option in CORR procedure 270-271
- scatter plots 236-237, 270-271
- SCATTER statement in SGPLOT procedure 236-237
- SCHEFFE option in ANOVA procedure 276-279
- Scheffe's multiple-comparisons 276-279
- scientific notation
 - format for writing 112-113
 - reading data with 40, 42
- SELECT statement in SQL procedure 326-328
- selecting observations
 - DELETE statements 86-87
 - IF statements 86-87
 - IN= data set option 200-201
 - INPUT statements 54-55
 - OUTPUT statement 194-195
 - reading raw data file 54-55
 - saving memory and disk space 323
 - WHERE statement 102-103, 328-330
 - WHERE= data set option 202-203, 329-330
- selecting output objects 153
- semicolon 2
 - missing 300-301
- sequential files 36
- series plots 238-239
- SERIES statement in SGPLOT procedure 238-239
- SERROR system option 224-225
- SET statement
 - BY statement 182-183
 - combining grand total with data 190-191
 - combining one observation with many 190-191
 - concatenating data sets 180-181
 - interleaving data sets 182-183
 - modifying single data set 178-179
 - multiple SET statements 190-191
 - reading single data set 178-179
 - stacking data sets 180-181
- SGPANEL procedure 228, 248-249
- SGPLOT procedure 228, 230-247
- sharing data with other software 30-31
- SHEET= statement in the IMPORT procedure 62
- SHOWBINS option in HISTOGRAM statement 232-233
- SIDES= option in TTEST procedure 260
- size
 - data sets 5, 70-71
 - footnotes 164-165

size (*continued*)

- graphics images 250
 - titles 164-165
 - variables 5, 24-25, 70-71, 318-319, 322-323
- SIZE= option for graph attributes 246-247
- skewness
- MEANS procedure 258
 - UNIVARIATE procedure 255
- SKEWNESS option in MEANS procedure 258
- skipping over variables at input 40-41
- Somer's D 264
- SORT procedure 104-107
- sorting data in Viewtable 23
- SORTSEQ= option in SORT procedure 106-107
- space-delimited raw data
- reading 34-35, 60-61
 - writing 284-287
- SPACING= option in PANELBY statement 248-249
- SPANROWS option in REPORT procedure 168-169
- Spearman coefficient 264, 268
- SPEARMAN option in CORR procedure 268
- splitting
- data file 54-55
 - SAS data set 194-195, 202-203, 329-330
- SPSS files
- data engine 31
 - reading 34
 - writing 284
- SQL compared to SAS 325-330
- SQL procedure 325-328
- STACKED option in FREQ procedure 266
- stacking SAS data sets 180-181
- standard deviation
- MEANS procedure 118-119, 258
 - REPORT procedure 142
 - TABULATE procedure 126
 - UNIVARIATE procedure 254-255
- standard error
- MEANS procedure 258
 - REG procedure 273
- STARTPAGE= option
- ODS PDF statement 162-163
 - ODS RTF statement 160-161
- STAT= option in VBAR statement 230
- Stata files
- reading 34
 - writing 284
- statement not valid error in log 314-315
- statement options
- compared to data set options 198-199
- statistics
- analysis of variance 276-279
 - categorical data 264-265
 - correlations 268-269
 - descriptive 118-143, 254-259
 - multiple comparisons 276-279
 - output data set, MEANS procedure 120-121
 - regression 272-273
 - t test 260-263, 276
- STD keyword in REPORT procedure 142
- STDDEV option
- MEANS procedure 118, 258
 - TABULATE procedure 126
- STDERR option in MEANS procedure 258
- STIMERw. informat 44-45
- STOP statement 7, 223
- STRENGTH= suboption in SORT procedure 106-107
- strings, character 4, 74-75
- Stuart's tau-c 264
- student's t 258
- style attributes
- for graphics 246-247
 - PRINT procedure 166-167
 - REPORT procedure 168-169
 - table of 174-175
 - TABULATE procedure 170-171
- style templates 150-151
- for graphics 229, 246-247, 251
- STYLE= option
- graph attributes 246
 - graphics 251
 - ODS HTML statement 158-159
 - ODS LISTING statement 229, 251

- ODS PDF statement 162-163
- ODS RTF statement 160-161
- PRINT procedure 166-167
- REPORT procedure 168-169
- TABULATE procedure 170-171
- traffic-lighting 172-173
- user-defined formats 172-173
- SUBMIT SAS windowing environment
 - command 14-15
- submitting SAS programs
 - methods 10-11
 - SAS windowing environment 14-15
- subsetting observations
 - DELETE statements 86-87
 - IF statements 86-87
 - IN= data set option 200-201
 - INPUT statements 54-55
 - OUTPUT statement 194-195
 - reading raw data file 54-55
 - saving memory and disk space 323
 - WHERE statement 102-103, 328-330
 - WHERE= data set option 202-203, 329-330
- SUBSTR function 78-79
- subtotals
 - PRINT procedure 108-109
 - REPORT procedure 140-141
- SUM function 80-81, 309
- SUM keyword
 - REPORT procedure 142
 - TABULATE procedure 126
- sum of squares
 - ANOVA procedure 278-279
 - MEANS procedure 258
 - REG procedure 272
- SUM option in MEANS procedure 118, 258
- SUM statement in PRINT procedure 108-109
- sum statements, DATA step 92-93
- SUMMARIZE option in REPORT procedure 140-141
- SUMMARY location in STYLE= option 168
- SUMMARY procedure 121
- summary statistics
 - MEANS procedure 118-119, 258-259
 - merging with original data 188-191
 - REPORT procedure 134-143
 - saving in SAS data set 120-121
 - TABULATE procedure 124-127
 - UNIVARIATE procedure 254-255
- SUMMARYPLOT option in TTEST procedure 262-263
- sums
 - across observations 92-93, 108-109, 118-121, 126-133
 - across variables 74-75, 80-81, 309
 - combining with data 188-191
 - controlling style in PRINT procedure 166
 - MEANS procedure 118-121
 - REPORT procedure 134-145
 - SUM function 80-81, 309
 - SUM keyword in TABULATE procedure 126
 - SUM option in MEANS procedure 118, 258
 - sum statement in DATA step 92-93
 - SUM statement in PRINT procedure 108-109
 - TABULATE procedure 126-133
- SUMWGT option in MEANS procedure 258
- SVG image format 250
- SYMBOL= option for graph attributes 246-247
- SYMBOLGEN system option 224-225
- SYMPUT, CALL 222-223
- syntax of SAS programs 2
- syntax-sensitive editor 12, 297
- syntax, checking 299
- system options 26-27
 - BOTTOMMARGIN= 27
 - CENTER/NOCENTER 27
 - compared to data set options 198-199
 - DATASTMTCHK= 301
 - DATE/NO DATE 27
 - LEFTMARGIN= 27
 - MACRO 211
 - MERROR 224-225
 - MLOGIC 224-225
 - MPRINT 224-225
 - NUMBER/NONNUMBER 27

system options (*continued*)

ORIENTATION= 27
 PAGENO= 27
 RIGHTMARGIN= 27
 SERROR 224-225
 SYMBOLGEN 224-225
 TOPMARGIN= 27
 VALIDVARNames= 5
 YEARCUTOFF= 88

T

T option

ANOVA procedure 276
 MEANS procedure 258

t tests

MEANS procedure 258
 ANOVA procedure 276
 TTEST procedure 260-263

TAB value in the DBMS= option

EXPORT procedure 286
 IMPORT procedure 60

tab-delimited data

reading 34-35, 58-61
 writing 284-287

Table Editor 32

TABLE statement in TABULATE procedure

124-133
 STYLE= option 170-171

table templates 150-151

tables of data

definition 4
 Viewtable window 32-33

TABLES statement in FREQ procedure 122-123, 264-265

TABULATE procedure 124-133

CLASSLEV statement 170
 STYLE= option 170-171

templates 150-151

temporary SAS data sets 64-65

text files

reading 36
 writing 150, 156-157

text, adding to graphs 244-245

THEN keyword 82-85

THICKNESS= option for graph attributes 246-247

TIFF image format 250

time data

formats 112-113
 informats 44-45

TIME option in XAXIS or YAXIS statement 242

TIMEw. informat 44-45

TIMEw.d format 112-113

title

default 39
 removing procedure name 156-163

TITLE statement 100-101, 164-165

TODAY function 80-81, 88-91

toolbar in SAS windowing environment 13

TOPMARGIN= system option 27

TOTAL location in STYLE= option 166

totals

across observations 92-93, 108-109, 118-121, 126-133

across variables 74-75, 80-81, 309

combining with data 188-191

controlling style in PRINT procedure 166

MEANS procedure 118-121

REPORT procedure 134-145

SUM function 80-81, 309

SUM keyword in TABULATE procedure 126

SUM option in MEANS procedure 118, 258

sum statement in DATA step 92-93

SUM statement in PRINT procedure 108-109

TABULATE procedure 126-133

tracing output objects 152-153

tracking observations

IN= data set option 200-201

traffic-lighting 172-173

trailing @ 54-55, 290

trailing @@ 52-53

transaction-oriented data 192-193

TRANSLATE function 78-79

TRANSPARENCY= option

- DENSITY statement 232
- HBOX or VBOX statement 234
- HISTOGRAM statement 232
- REFLINE statement 242-243
- SCATTER statement 236
- SERIES statement 238
- VBAR statement 230
- TRANPOSE procedure 204-205
- transposing data with OUTPUT statement 196-197
- TRANWRD function 78-79
- TREND option in FREQ procedure 264
- TRIM function 78-79
- truncation of character data 318-319
- TRUNCOVER option on INFILE statement 57, 303
- TTEST procedure 260-263
- TUKEY option in ANOVA procedure 276
- Tukey's studentized range test 276
- two-way frequency table 122-123, 264-265
- TWOWAY= option in FREQ procedure 266-267
- type of variable 4, 71
- TYPE= option
 - DENSITY statement 232-233
 - XAXIS or YAXIS statement 242-243

U

- UCLM option in MEANS procedure 258
- uninitialized variables 316-317
- UNISCALE= option in PANELBY statement 248
- UNIVARIATE procedure 254-257
- UNIX
 - direct referencing of SAS data sets 68
 - INFILE statement 37
 - LIBNAME statement 66
- UPCASE function 78-79
- UPDATE statement 192-193
- URL style attribute 174-175
- usage options in REPORT procedure 136-137
- user-defined formats 114-115
 - grouping with 146-147
 - traffic-lighting 172-173
 - with TABULATE procedure 130-131

- USS option in MEANS procedure 258

V

- VALIDVARNames= system option 5
- VALUE statement FORMAT procedure 114-115
- VALUEATTRS= option for graph attributes 246
- VALUES= option in XAXIS or YAXIS statement 242
- VAR option in MEANS procedure 258
- VAR statement
 - CORR procedure 268-269
 - MEANS procedure 118-119, 258-259
 - PRINT procedure 108-109
 - STYLE = option in TABULATE procedure 170
 - STYLE= option in PRINT procedure 166-167
 - TABULATE procedure 126-133
 - TRANPOSE procedure 204-205
 - TTEST procedure 260
 - UNIVARIATE procedure 254
- variable length records, reading 57
- variable length values, reading 48-49
- variable name lists
 - _ALL_ 96-97, 312-313
 - _CHARACTER_ 96-97
 - _NUMERIC_ 96-97
 - name prefix 96
 - name ranges 96-97
 - numbered ranges 96-97
- variable not found error in log 316-317
- variable uninitialized note in log 316-317
- variables
 - arrays 94-95
 - automatic 206-207
 - automatic macro 214-215
 - changing to observations 204-205
 - creating a grouping variable 84-85
 - creating in REPORT procedure 144-145
 - creating with assignment statements 74-75
 - definition 4
 - dropping 198-199
 - keeping 198-199
 - labels 70-71, 101

variables (*continued*)

- length 70-71, 318-319, 322-323
- lists 96-97
- means 118-119
- names 5
- printing 108-109
- renaming 198-199
- retaining values between observations 92-93
- skipping when reading raw data 40
- type 4, 71
- uninitialized 316-317
- variance with MEANS procedure 258
- VBAR statement in SGLOT procedure 230-231
- VBOX statement 234-235
- vector graphics, scalable 250
- views with SQL procedure 325-326
- Viewtable window 22-23, 32-33

W

- w.d* format 112-113
- w.d* informat 44-45
- warnings in SAS log 298
- Web, creating files for 158-159
- WEEKDATE*w.* format 90-91, 112-113
- WEEKDAY function 80-81, 90-91
- WEIBULL option in distribution plots 256
- WEIGHT= option for graph attributes 246-247
- WHERE statement
 - compared to subsetting IF 328-330
 - DATA steps 328-330
 - procedures 102-103, 328-330
- WHERE= data set option 198, 202-203, 329-330
- WIDTH= option in ODS GRAPHICS statement 250-251
- windowing environment, SAS 12-25, 156-157
- Windows operating environment
 - direct referencing of SAS data sets 68
 - INFILE statement 37
 - LIBNAME statement 66
- WITH statement in CORR procedure 268-269
- Wizard
 - Export 284-285
 - Import 34-35

- WORDDATE*w.* format 90-91, 112-113
- WORK library 20-21, 64-65
- writing data 282-293
 - delimited 284-287, 290-293
 - methods 282-283
 - PC files 282-289
 - raw data 284-287, 290-293
- writing SAS data sets
 - DATA step 6-7
 - multiple data sets 194-195
 - permanent data sets 64-69
- WTKAPPAPLOT option in FREQ procedure 266

X

- XAXIS statement 242-243
- XML output 150, 282

Y

- YAXIS statement 242-243
- YEAR function 80-81, 90-91
- YEARCUTOFF= system option 88
- YRDIF function 80-81, 88-91

Z

- z/OS
 - comments 321
 - direct referencing of SAS data sets 68
 - INFILE statement 37
 - LIBNAME statement 66

Special Characters

- ^= comparison operator 82, 102
- _ALL_ variable name list 96
 - in PUT statements 312-313
- _CHARACTER_ variable name list 96-97
- _ERROR_ automatic variable 206
 - invalid data message 306-307
- _FREQ_ variable in MEANS procedure 120-121
- _N_ automatic variable 206-207
 - invalid data message 306-307

- _NAME_ variable
 - TRANSPOSE procedure 204-205
- _NULL_ data set name 116-117, 290-291
- _NUMERIC_ variable name list 96-97
- _PAGE_ keyword in PUT statements 116-117
- _TYPE_ variable, MEANS procedure 120-121
- ; semicolon 2
 - missing 300-301
- : colon modifier 48-49
- ! comparison operator 83, 102
- ?? informat modifier 307
- @ line-hold specifier 54-55, 290
 - compared to @@ 55
- @'character' column pointer 48-49
- @@ line-hold specifier 52-53
 - compared to @ 55
- @n column pointer 46-47, 290-291
- * ; comments 3
- / line pointer 50-51, 290
- /* */ comments 3
 - in z/OS 320
- & comparison operator 83, 102
- & macro variable prefix 210
- &SYSDATE macro variable 214-215
- &SYSDAY macro variable 214
- &SYSNOBS macro variable 214
- #n line pointer 50-51, 290
- % macro prefix 210
- %DO statements 220-221
- %ELSE statement 220-221
- %END statement 220-221
- %IF-%THEN statements 220-221
- %LET statement 212-215
- %MACRO statement 216-219
- %MEND statement 216-217
- +n column pointer 42-43
- < comparison operator 82-83, 102
- <= comparison operator 82, 102
- = comparison operator 82-83, 102-103
- > comparison operator 82-83, 102
- >= comparison operator 82, 102
- ¬= comparison operator 82, 102
- | comparison operator 83, 102
- || concatenation operator 79
- ~= comparison operator 82, 102
- \$CHARw. informat 44-45
- \$UPCASEw. format 112-113
- \$UPCASEw. informat 44-45
- \$w. format 112-113
- \$w. informat 44-45

About These Authors



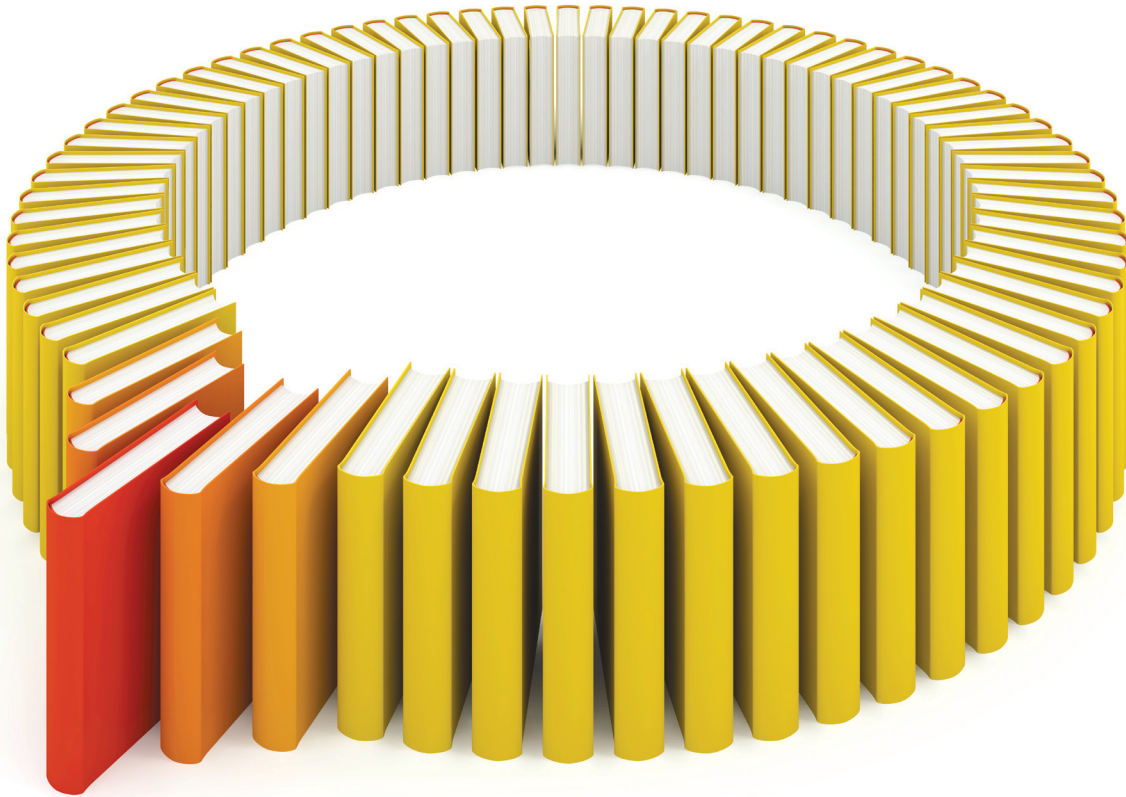
With over 25 years of experience, Lora D. Delwiche enjoys teaching people about SAS software and likes solving challenging problems using SAS. She has spent most of her career at the University of California, Davis, using SAS in support of teaching and research.

Susan J. Slaughter discovered SAS software in graduate school over 25 years ago. Since then, she has used SAS in a variety of business and academic settings. She now works as a consultant through her company, Avocet Solutions.

Learn more about these authors by visiting their author pages, where you can download free chapters, access example code and data, read the latest reviews, get updates, and more:

<http://support.sas.com/delwiche>

<http://support.sas.com/slaughter>



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

