



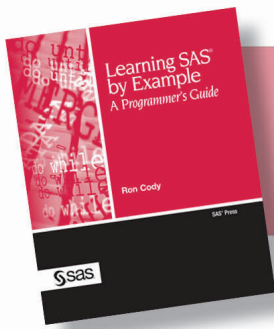
Learning SAS[®] by Example

A Programmer's Guide

Ron Cody

SAS[®] Press





From *Learning SAS® by Example*. Full book available for purchase [here](#).

Contents

List of Programs xv

Preface xxix

Acknowledgments xxxi

Part 1 Getting Started 1

Chapter 1 What Is SAS? 3

- 1.1 Introduction 3
- 1.2 Getting Data into SAS 4
- 1.3 A Sample SAS Program 4
- 1.4 SAS Names 7
- 1.5 SAS Data Sets and SAS Data Types 8
- 1.6 The SAS Display Manager and SAS Enterprise Guide 9
- 1.7 Problems 9

Chapter 2 Writing Your First SAS Program 11

- 2.1 A Simple Program to Read Raw Data and Produce a Report 11
- 2.2 Enhancing the Program 18
- 2.3 More on Comment Statements 20
- 2.4 How SAS Works (a Look Inside the “Black Box”) 22
- 2.5 Problems 25

Part 2 DATA Step Processing 27

Chapter 3 Reading Raw Data from External Files 29

- 3.1 Introduction 30
- 3.2 Reading Data Values Separated by Blanks 30
- 3.3 Specifying Missing Values with List Input 32
- 3.4 Reading Data Values Separated by Commas (CSV Files) 33
- 3.5 Using an Alternative Method to Specify an External File 34

- 3.6 Reading Data Values Separated by Delimiters Other Than Blanks or Commas 34
- 3.7 Placing Data Lines Directly in Your Program (the DATALINES Statement) 36
- 3.8 Specifying INFILE Options with the DATALINES Statement 37
- 3.9 Reading Raw Data from Fixed Columns—Method 1: Column Input 37
- 3.10 Reading Raw Data from Fixed Columns—Method 2: Formatted Input 39
- 3.11 Using a FORMAT Statement in a DATA Step versus in a Procedure 43
- 3.12 Using Informats with List Input 43
- 3.13 Supplying an INFORMAT Statement with List Input 45
- 3.14 Using List Input with Embedded Delimiters 46
- 3.15 Problems 47

Chapter 4 Creating Permanent SAS Data Sets 53

- 4.1 Introduction 54
- 4.2 SAS Libraries—The LIBNAME Statement 54
- 4.3 Why Create Permanent SAS Data Sets? 55
- 4.4 Examining the Descriptor Portion of a SAS Data Set Using PROC CONTENTS 56
- 4.5 Listing All the SAS Data Sets in a SAS Library Using PROC CONTENTS 59
- 4.6 Viewing the Descriptor Portion of a SAS Data Set Using the SAS Explorer 60
- 4.7 Viewing the Data Portion of a SAS Data Set Using PROC PRINT 63
- 4.8 Viewing the Data Portion of a SAS Data Set Using the SAS VIEWTABLE Window 64
- 4.9 Using a SAS Data Set as Input to a DATA Step 65
- 4.10 DATA _NULL_: A Data Set That Isn't 67
- 4.11 Problems 68

Chapter 5 Creating Formats and Labels 71

- 5.1 Adding Labels to Your Variables 71
- 5.2 Using Formats to Enhance Your Output 73
- 5.3 Regrouping Values Using Formats 76
- 5.4 More on Format Ranges 78
- 5.5 Storing Your Formats in a Format Library 79
- 5.6 Permanent Data Set Attributes 80
- 5.7 Accessing a Permanent SAS Data Set with User-Defined Formats 82
- 5.8 Displaying Your Format Definitions 83
- 5.9 Problems 84

Chapter 6 Reading and Writing Data from an Excel Spreadsheet 87

- 6.1 Introduction 87
- 6.2 Using the Import Wizard to Convert a Spreadsheet to a SAS Data Set 88
- 6.3 Creating an Excel Spreadsheet from a SAS Data Set 93
- 6.4 Using an Engine to Read an Excel Spreadsheet 95
- 6.5 Using the SAS Output Delivery System to Convert a SAS Data Set to an Excel Spreadsheet 96
- 6.6 Problems 98

Chapter 7 Performing Conditional Processing 101

- 7.1 Introduction 102
- 7.2 The IF and ELSE IF Statements 102
- 7.3 The Subsetting IF Statement 105
- 7.4 The IN Operator 107
- 7.5 Using a SELECT Statement for Logical Tests 108
- 7.6 Using Boolean Logic (AND, OR, and NOT Operators) 109
- 7.7 A Caution When Using Multiple OR Operators 111
- 7.8 The WHERE Statement 112
- 7.9 Some Useful WHERE Operators 113
- 7.10 Problems 114

Chapter 8 Performing Iterative Processing: Looping 117

- 8.1 Introduction 117
- 8.2 DO Groups 118
- 8.3 The Sum Statement 120
- 8.4 The Iterative DO Loop 125
- 8.5 Other Forms of an Iterative DO Loop 129
- 8.6 DO WHILE and DO UNTIL Statements 131
- 8.7 A Caution When Using DO UNTIL Statements 134
- 8.8 LEAVE and CONTINUE Statements 135
- 8.9 Problems 137

Chapter 9 Working with Dates 141

- 9.1 Introduction 142
- 9.2 How SAS Stores Dates 142
- 9.3 Reading Date Values from Raw Data 143
- 9.4 Computing the Number of Years between Two Dates 146
- 9.5 Demonstrating a Date Constant 147
- 9.6 Computing the Current Date 148
- 9.7 Extracting the Day of the Week, Day of the Month, Month, and Year from a SAS Date 149
- 9.8 Creating a SAS Date from Month, Day, and Year Values 150
- 9.9 Substituting the 15th of the Month when the Day Value Is Missing 151
- 9.10 Using Date Interval Functions 152
- 9.11 Problems 157

Chapter 10 Subsetting and Combining SAS Data Sets 161

- 10.1 Introduction 162
- 10.2 Subsetting a SAS Data Set 162
- 10.3 Creating More Than One Subset Data Set in One DATA Step 163
- 10.4 Adding Observations to a SAS Data Set 164
- 10.5 Interleaving Data Sets 167
- 10.6 Combining Detail and Summary Data 168

- 10.7 Merging Two Data Sets 170
- 10.8 Omitting the BY Statement in a Merge 172
- 10.9 Controlling Observations in a Merged Data Set 173
- 10.10 More Uses for IN= Variables 175
- 10.11 When Does a DATA Step End? 176
- 10.12 Merging Two Data Sets with Different BY Variable Names 177
- 10.13 Merging Two Data Sets with Different BY Variable Data Types 179
- 10.14 One-to-One, One-to-Many, and Many-to-Many Merges 181
- 10.15 Updating a Master File from a Transaction File 183
- 10.16 Problems 185

Chapter 11 Working with Numeric Functions 189

- 11.1 Introduction 190
- 11.2 Functions That Round and Truncate Numeric Values 190
- 11.3 Functions That Work with Missing Values 192
- 11.4 Setting Character and Numeric Values to Missing 193
- 11.5 Descriptive Statistics Functions 194
- 11.6 Computing Sums within an Observation 196
- 11.7 Mathematical Functions 197
- 11.8 Computing Some Useful Constants 198
- 11.9 Generating Random Numbers 199
- 11.10 Special Functions 201
- 11.11 Functions That Return Values from Previous Observations 204
- 11.12 Problems 207

Chapter 12 Working with Character Functions 211

- 12.1 Introduction 212
- 12.2 Determining the Length of a Character Value 212
- 12.3 Changing the Case of Characters 213
- 12.4 Removing Characters from Strings 214
- 12.5 Joining Two or More Strings Together 215
- 12.6 Removing Leading or Trailing Blanks 217

- 12.7 Using the COMPRESS Function to Remove Characters from a String 218
- 12.8 Searching for Characters 220
- 12.9 Searching for Individual Characters 223
- 12.10 Searching for Words in a String 223
- 12.11 Searching for Character Classes 225
- 12.12 Using the NOT Functions for Data Cleaning 226
- 12.13 Describing a Real Blockbuster Data Cleaning Function 227
- 12.14 Extracting Part of a String 228
- 12.15 Dividing Strings into Words 230
- 12.16 Comparing Strings 232
- 12.17 Performing a Fuzzy Match 234
- 12.18 Substituting Characters or Words 235
- 12.19 Problems 238

Chapter 13 Working with Arrays 243

- 13.1 Introduction 244
- 13.2 Setting Values of 999 to a SAS Missing Value for Several Numeric Variables 244
- 13.3 Setting Values of NA and ? to a Missing Character Value 247
- 13.4 Converting All Character Values to Lowercase 248
- 13.5 Using an Array to Create New Variables 249
- 13.6 Changing the Array Bounds 250
- 13.7 Temporary Arrays 251
- 13.8 Loading the Initial Values of a Temporary Array from a Raw Data File 253
- 13.9 Using a Multidimensional Array for Table Lookup 254
- 13.10 Problems 257

Part 3 Presenting and Summarizing Your Data 259**Chapter 14 Displaying Your Data 261**

- 14.1 Introduction 262
- 14.2 The Basics 262
- 14.3 Changing the Appearance of Your Listing 263
- 14.4 Changing the Appearance of Values 265
- 14.5 Controlling the Observations That Appear in Your Listing 266
- 14.6 Adding Additional Titles and Footnotes to Your Listing 268
- 14.7 Changing the Order of Your Listing 270
- 14.8 Sorting by More Than One Variable 272
- 14.9 Labeling Your Column Headings 273
- 14.10 Adding Subtotals and Totals to Your Listing 274
- 14.11 Making Your Listing Easier to Read 277
- 14.12 Adding the Number of Observations to Your Listing 279
- 14.13 Double-Spacing Your Listing 280
- 14.14 Listing the First n Observations of Your Data Set 281
- 14.15 Problems 283

Chapter 15 Creating Customized Reports 287

- 15.1 Introduction 288
- 15.2 Using PROC REPORT 289
- 15.3 Selecting Variables to Include in Your Report 291
- 15.4 Comparing Detail and Summary Reports 291
- 15.5 Producing a Summary Report 293
- 15.6 Demonstrating the FLOW Option of PROC REPORT 294
- 15.7 Using Two Grouping Variables 296
- 15.8 Changing the Order of Variables in the COLUMN Statement 297
- 15.9 Changing the Order of Rows in a Report 299
- 15.10 Applying the ORDER Usage to Two Variables 300
- 15.11 Creating a Multi-Column Report 301
- 15.12 Producing Report Breaks 303
- 15.13 Using a Nonprinting Variable to Order a Report 306
- 15.14 Computing a New Variable with PROC REPORT 307
- 15.15 Computing a Character Variable in a COMPUTE Block 308

- 15.16 Creating an ACROSS Variable with PROC REPORT 310
- 15.17 Modifying the Column Label for an ACROSS Variable 311
- 15.18 Using an ACROSS Usage to Display Statistics 311
- 15.19 Problems 313

Chapter 16 Summarizing Your Data 319

- 16.1 Introduction 320
- 16.2 PROC MEANS—Starting from the Beginning 320
- 16.3 Adding a BY Statement to PROC MEANS 323
- 16.4 Using a CLASS Statement with PROC MEANS 324
- 16.5 Applying a Format to a CLASS Variable 325
- 16.6 Deciding between a BY Statement and a CLASS Statement 327
- 16.7 Creating Summary Data Sets Using PROC MEANS 327
- 16.8 Outputting Other Descriptive Statistics with PROC MEANS 328
- 16.9 Asking SAS to Name the Variables in the Output Data Set 329
- 16.10 Outputting a Summary Data Set: Including a BY Statement 330
- 16.11 Outputting a Summary Data Set: Including a CLASS Statement 331
- 16.12 Using Two CLASS Variables with PROC MEANS 333
- 16.13 Selecting Different Statistics for Each Variable 337
- 16.14 Problems 338

Chapter 17 Counting Frequencies 341

- 17.1 Introduction 342
- 17.2 Counting Frequencies 342
- 17.3 Selecting Variables for PROC FREQ 345
- 17.4 Using Formats to Label the Output 346
- 17.5 Using Formats to Group Values 347
- 17.6 Problems Grouping Values with PROC FREQ 349
- 17.7 Displaying Missing Values in the Frequency Table 351
- 17.8 Changing the Order of Values in PROC FREQ 353
- 17.9 Producing Two-Way Tables 356

- 17.10 Requesting Multiple Two-Way Tables 358
- 17.11 Producing Three-Way Tables 358
- 17.12 Problems 360

Chapter 18 Creating Tabular Reports 363

- 18.1 Introduction 364
- 18.2 A Simple PROC TABULATE Table 364
- 18.3 Describing the Three PROC TABULATE Operators 366
- 18.4 Using the Keyword ALL 369
- 18.5 Producing Descriptive Statistics 370
- 18.6 Combining CLASS and Analysis Variables in a Table 372
- 18.7 Customizing Your Table 374
- 18.8 Demonstrating a More Complex Table 377
- 18.9 Computing Row and Column Percentages 379
- 18.10 Displaying Percentages in a Two-Dimensional Table 381
- 18.11 Computing Column Percentages 382
- 18.12 Computing Percentages on Numeric Variables 384
- 18.13 Understanding How Missing Values Affect PROC TABULATE Output 385
- 18.14 Problems 390

Chapter 19 Introducing the Output Delivery System 397

- 19.1 Introduction 397
- 19.2 Sending SAS Output to an HTML File 398
- 19.3 Creating a Table of Contents 400
- 19.4 Selecting a Different HTML Style 401
- 19.5 Choosing Other ODS Destinations 402
- 19.6 Selecting or Excluding Portions of SAS Output 403
- 19.7 Sending Output to a SAS Data Set 407
- 19.8 Problems 409

Chapter 20 Generating High-Quality Graphics 411

- 20.1 Introduction 412
- 20.2 Some Basic Concepts 412
- 20.3 Producing Simple Bar Charts Using PROC GCHART 413
- 20.4 Creating Pie Charts 415
- 20.5 Creating Bar Charts for a Continuous Variable 416

- 20.6 Creating Charts with Values Representing Categories 418
- 20.7 Creating Bar Charts Representing Sums 420
- 20.8 Creating Bar Charts Representing Means 422
- 20.9 Adding Another Variable to the Chart 423
- 20.10 Producing Scatter Plots 425
- 20.11 Connecting Points 427
- 20.12 Connecting Points with a Smooth Line 430
- 20.13 Problems 431

Part 4 Advanced Topics 435

Chapter 21 Using Advanced INPUT Techniques 437

- 21.1 Introduction 438
- 21.2 Handling Missing Values at the End of a Line 438
- 21.3 Reading Short Data Lines 440
- 21.4 Reading External Files with Lines Longer Than 256 Characters 443
- 21.5 Detecting the End of the File 443
- 21.6 Reading a Portion of a Raw Data File 445
- 21.7 Reading Data from Multiple Files 446
- 21.8 Reading Data from Multiple Files Using a FILENAME Statement 447
- 21.9 Reading External Filenames from a Data File 447
- 21.10 Reading Multiple Lines of Data to Form One Observation 448
- 21.11 Reading Data Conditionally (the Single Trailing @ Sign) 451
- 21.12 More Examples of the Single Trailing @ Sign 453
- 21.13 Creating Multiple Observations from One Line of Input 454
- 21.14 Using Variable and Informat Lists 455
- 21.15 Using Relative Column Pointers to Read a Complex Data Structure Efficiently 456
- 21.16 Problems 458

Chapter 22 Using Advanced Features of User-Defined Formats and Informats 462

- 22.1 Introduction 462
- 22.2 Using Formats to Recode Variables 462
- 22.3 Using Formats with a PUT Function to Create New Variables 463
- 22.4 Creating User-Defined Informats 464
- 22.5 Reading Character and Numeric Data in One Step 467
- 22.6 Using Formats (and Informats) to Perform Table Lookup 470
- 22.7 Using a SAS Data Set to Create a Format 471
- 22.8 Updating and Maintaining Your Formats 477
- 22.9 Using Formats within Formats 479
- 22.10 Using Multilabel Formats 482
- 22.11 Using the INPUTN Function to Perform a More Complicated Table Lookup 485
- 22.12 Problems 490

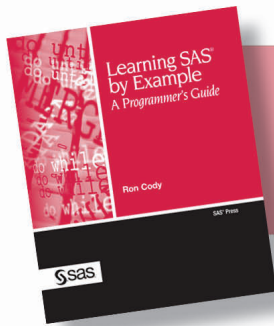
Chapter 23 Restructuring SAS Data Sets 493

- 23.1 Introduction 494
- 23.2 Converting a Data Set with One Observation per Subject to a Data Set with Several Observations per Subject: Using a DATA Step 494
- 23.3 Converting a Data Set with Several Observations per Subject to a Data Set with One Observation per Subject: Using a DATA Step 496
- 23.4 Converting a Data Set with One Observation per Subject to a Data Set with Several Observations per Subject: Using PROC TRANSPOSE 498
- 23.5 Converting a Data Set with Several Observations per Subject to a Data Set with One Observation per Subject: Using PROC TRANSPOSE 500
- 23.6 Problems 501

Chapter 24 Working with Multiple Observations per Subject 505

- 24.1 Introduction 506
- 24.2 Identifying the First or Last Observation in a Group 506
- 24.3 Counting the Number of Visits Using PROC FREQ 509

24.4	Counting the Number of Visits Using PROC MEANS	511
24.5	Computing Differences between Observations	512
24.6	Computing Differences between the First and Last Observation in a BY Group Using the LAG Function	514
24.7	Computing Differences between the First and Last Observation in a BY Group Using a RETAIN Statement	515
24.8	Using a Retained Variable to “Remember” a Previous Value	517
24.9	Problems	518
Chapter 25	Introducing the SAS Macro Language	521
25.1	Introduction	522
25.2	Macro Variables: What Are They?	522
25.3	Some Built-In Macro Variables	523
25.4	Assigning Values to Macro Variables with a %LET Statement	524
25.5	Demonstrating a Simple Macro	525
25.6	A Word about Tokens	527
25.7	Another Example of Using a Macro Variable as a Prefix	529
25.8	Using a Macro Variable to Transfer a Value between DATA Steps	530
25.9	Problems	532
Chapter 26	Introducing the Structured Query Language	535
26.1	Introduction	536
26.2	Some Basics	536
26.3	Joining Two Tables (Merge)	539
26.4	Left, Right, and Full Joins	543
26.5	Concatenating Data Sets	546
26.6	Using Summary Functions	549
26.7	Demonstrating an ORDER Clause	551
26.8	An Example of Fuzzy Matching	551
26.9	Problems	553
	Solutions to Odd-Numbered Problems	557
	Index	601



From *Learning SAS® by Example*. Full book available for purchase [here](#).



Chapter 1

What Is SAS?

- 1.1 Introduction 3
- 1.2 Getting Data into SAS 4
- 1.3 A Sample SAS Program 4
- 1.4 SAS Names 7
- 1.5 SAS Data Sets and SAS Data Types 8
- 1.6 The SAS Display Manager and SAS Enterprise Guide 9
- 1.7 Problems 9

1.1 Introduction

SAS is a collection of modules that are used to process and analyze data. It began in the late '60s and early '70s as a statistical package (the name SAS originally stood for Statistical Analysis System). However, unlike many competing statistical packages, SAS is also an extremely powerful, general-purpose programming language. We see SAS as the predominant software in the pharmaceutical industry and most Fortune 500

4 *Learning SAS by Example: A Programmer's Guide*

companies. In recent years, it has been enhanced to provide state-of-the-art data mining tools and programs for Web development and analysis.

This book covers most of the basic data management and programming tools provided in Base SAS. Statistical procedures are not covered here.¹

The only way to really learn a programming language is to write lots of programs, make some errors, correct the errors, and then make some more. You can download all the programs and data files used in this book from this book's companion Web site at <http://support.sas.com/cody> and from the CD that accompanies this book. If you already have access to SAS at work or school, you are ready to go. If you are learning SAS on your own and do not have a copy of SAS to play with, we highly recommend that you obtain the SAS Learning Edition 4.1. This is a relatively inexpensive, fully functional version of SAS that was developed primarily for students for learning purposes only. Anyone can buy it, either through SAS Publishing, Amazon.com, or other retailers. With a pre-set die date of 12/31/08, you can use the SAS Enterprise Guide 4.1 point-and-click interface, or write and modify SAS code using the SAS Program Editor. You will be able to run any program in this book using the SAS Learning Edition...it is an ideal way to learn SAS.

1.2 Getting Data into SAS

SAS can read data from almost any source. Common sources of data are raw text files, Microsoft Office Excel spreadsheets, Access databases, and most of the common database systems such as DB2 and Oracle. Most of this book uses either text files or Excel spreadsheets as data sources.

1.3 A Sample SAS Program

Let's start out with a simple SAS program that reads data from a text file and produces some basic reports to give you an overview of the structure of SAS programs.

¹ See Ron Cody and Jeffrey K. Smith, *Applied Statistics and the Programming Language, 5th ed.* (Englewood Cliffs, NJ: Prentice Hall, 2005), which is available from SAS Press, for details on using SAS for statistical analysis.

For this example, we have a text file with data on vegetable seeds. Each line of the file contains the following pieces of information (separated by spaces):

- Vegetable name
- Product code
- Days to germination
- Number of seeds
- Price

In SAS terminology, each piece of information is called a *variable*. (Other database systems, and sometimes SAS, use the term *column*.) A few sample lines from the file are shown here:

File `c:\books\learning\veggies.txt`

Cucumber	50104-A	55	30	195
Cucumber	51789-A	56	30	225
Carrot	50179-A	68	1500	395
Carrot	50872-A	65	1500	225
Corn	57224-A	75	200	295
Corn	62471-A	80	200	395
Corn	57828-A	66	200	295
Eggplant	52233-A	70	30	225

In this example, each line of data produces what SAS calls an *observation* (also referred to as a *row* in other systems). A complete SAS program to read this data file and produce a list of the data, a frequency count showing the number of entries for each vegetable, the average price per seed, and the average number of days until germination is shown here:

Program 1-1 A sample SAS program

```
*SAS Program to read veggie data file and to produce
several reports;

options nocenter nonumber;

data veg;
  infile "c:\books\learning\veggies.txt";
  input Name $ Code $ Days Number Price;
  CostPerSeed = Price / Number;
run;
```


6 Learning SAS by Example: A Programmer's Guide

```
{ title "List of the Raw Data";  
  proc print data=veg;  
  run;  
  
{ title "Frequency Distribution of Vegetable Names";  
  proc freq data=veg;  
    tables Name;  
  run;  
  
{ title "Average Cost of Seeds";  
  proc means data=veg;  
    var Price Days;  
  run;
```

At this point in the book, we won't explain every line of the program—we'll just give an overview.

SAS programs often contain DATA steps and PROC steps. *DATA steps* are parts of the program where you can read or write the data, manipulate the data, and perform calculations. *PROC* (short for procedure) *steps* are parts of your program where you ask SAS to run one or more of its procedures to produce reports, summarize the data, generate graphs, and much more. DATA steps begin with the word DATA and PROC steps begin with the word PROC. Most DATA and PROC steps end with a RUN statement (more on this later). SAS processes each DATA or PROC step completely and then goes on to the next step.

SAS also contains *global* statements that affect the entire SAS environment and remain in effect from one DATA or PROC step to another. In the program above, the OPTIONS and TITLE statements are examples of global statements. It is important to keep in mind that the actions of global statements remain in effect until they are changed by another global statement or until you end your SAS session.

All SAS programs, whether part of DATA or PROC steps, are made up of statements. Here is the rule: all SAS statements end with semicolons. This is an important rule because if you leave out a semicolon where one is needed, the program may not run correctly, resulting in hard-to-interpret error messages.

Let's discuss some of the basic rules of SAS statements. First, they can begin in any column and can span several lines, if necessary. Because a semicolon determines the end of a SAS statement, you can place more than one statement on a single line (although this is not recommended as a matter of style).

To help make this clear, let's look at some of the statements in Program 1-1.

You could write the DATA step as shown in Program 1-2. Although this program is identical to the original, notice that it doesn't look organized, making it hard to read. Notice, too, that spacing is not critical either, though it is useful for legibility. It is a common practice to start each SAS statement on a new line and to indent each statement within a DATA or PROC step by several spaces (this author likes three spaces).

Program 1-2 An alternative version of Program 1-1

```
data veg; infile "c:\books\learning\veggies.txt"; input
Name $ Code $ Days Number
Price; CostPerSeed =
Price /
Number;
run;
```

Another thing to notice about this program is that SAS is not case sensitive. Well, this is almost true. Of course references to external files must match the rules of your particular operating system. So, if you are running SAS under UNIX or Linux, file names will be case-sensitive. As you will see later, you get to name the variables in a SAS data set. The variable names in Program 1-1 are Name, Code, Days, Number, Price, and CostPerSeed. Although SAS doesn't care whether you write these names in uppercase, lowercase, or mixed case, it does "remember" the case of each variable the *first* time it encounters that variable and uses that form of the variable name when producing printed reports.

1.4 SAS Names

SAS names follow a simple naming rule: All SAS variable names and data set names can be no longer than 32 characters and must begin with a letter or the underscore (_) character. The remaining characters in the name may be letters, digits, or the underscore character. Characters such as dashes and spaces are not allowed. Here are some valid and invalid SAS names.

Valid SAS Names

Parts
LastName
First_Name
Ques5
Cost_per_Pound
DATE
time
X12Y34Z56

Invalid SAS Names

8_is_enough	Begins with a number
Price per Pound	Contains blanks
Month-total	Contains an invalid character (-)
Num%	Contains an invalid character (%)

1.5 SAS Data Sets and SAS Data Types

We will talk a lot about SAS data sets throughout this book. For now, you need to know that when SAS reads data from anywhere (for example, raw data, spreadsheets), it stores the data in its own special form called a SAS data set. Only SAS can read and write SAS data sets. If you opened a SAS data set with another program (Microsoft Word, for example), it would not be a pretty sight—it would consist of some recognizable characters and many funny-looking graphics characters. In other words, it would look like nonsense. Even if SAS is reading data from Oracle tables or DB2, it is actually converting the data into SAS data set format in the background.

The good news is that you don't ever have to worry about how SAS is storing its data or the structure of a SAS data set. However, it is important to understand that SAS data sets contain two parts: a descriptor portion and a data portion. Not only does SAS store the actual data values for you, it stores information about these values (things like storage lengths, labels, and formats). We'll discuss that more later.

SAS has only two types of variables: character and numeric. This makes it much simpler to use and understand than some other programs that have many more data types (for example, integer, long integer, and logical). SAS determines a fixed storage length for every variable. Most SAS users never need to think about storage lengths for numerical

values—they are stored in 8 bytes (about 14 or 15 significant digits, depending on your operating system) if you don't specify otherwise. The majority of SAS users will never have to change this default value (it can lead to complications and should only be considered by experienced SAS programmers). Each character value (data stored as letters, special characters, and numerals) is assigned a fixed storage length explicitly by program statements or by various rules that SAS has about the length of character values.

1.6 The SAS Display Manager and SAS Enterprise Guide

Because SAS runs on many different platforms (mainframes, microcomputers running various Microsoft operating systems, UNIX, and Linux), the way you write and run programs will vary. You might use a general-purpose text editor on a mainframe to write a SAS program, submit it, and send the output back to a terminal or to a file. On PCs, you might use the SAS Display Manager, where you write your program in the Enhanced Editor (Editor window), see any error messages and comments about your program and the data in the Log window, and view your output in the Output window. In addition to the Enhanced Editor, an older program, simply called the Program Editor, is available for Windows and UNIX users. As an alternative to the Display Manager, you may enter the SAS environment using SAS Enterprise Guide, which is a front-end to SAS that allows you to use a menu-driven system to write SAS programs and produce reports.

There are many excellent books published by SAS that offer detailed instructions on how to run SAS programs on each specific platform and the appropriate access method into SAS. This book concentrates on how to write SAS programs. You will find that SAS programs, regardless of what computer or operating system you are using, look basically the same. Typically, the only changes you need to make to migrate a SAS program from one platform to another is the way you describe external data sources and where you store SAS programs and output.

1.7 Problems

Solutions to odd-numbered problems are located at the back of this book and on the CD that accompanies this book. Solutions to all problems are available to professors. If you are a professor, visit the book's companion Web site at <http://support.sas.com/cody> for information about how to obtain the solutions to all problems.

10 *Learning SAS by Example: A Programmer's Guide*

1. Identify which of the following variable names are valid SAS names:

Height
HeightInCentimeters
Height_in_centimeters
Wt-Kg
x123y456
76Trombones
MiXeDCasE

2. In the following list, classify each data set name as valid or invalid:

Clinic
clinic
work
hyphens-in-the-name
123GO
Demographics_2006

3. You have a data set consisting of Student ID, English, History, Math, and Science test scores on 10 students.
 - a. The number of variables is _____
 - b. The number of observations is _____
4. True or false:
 - a. You can place more than one SAS statement on a single line.
 - b. You can use several lines for a single SAS statement.
 - c. SAS has three data types: character, numeric, and integer.
 - d. OPTIONS and TITLE statements are considered global statements.
5. What is the default storage length for SAS numeric variables (in bytes)?



From *Learning SAS® by Example*. Full book available for purchase [here](#).

Index

A

- ABS function 197–198
- absolute column pointer 456–457
- ACROSS option, DEFINE statement (REPORT)
 - creating ACROSS variable 310
 - displaying statistics 311–313
 - modifying column label 311
- addition in assignment statements 19–20
- addresses, standardizing 236–238
- AFTER option, RBREAK statement (REPORT) 303
- alignment parameter 156
- _ALL_ keyword 59, 369, 374–375
- ampersand (&) 46
- ANALYSIS option, DEFINE statement (REPORT) 292–295, 312
- analysis variables
 - DEFINE statement (REPORT) 292–295, 312
 - TABULATE procedure and 372–373, 377–378
- AND operator 109–111
- ANOVA procedure 463
- ANY functions 225–226
- ANYALNUM function 225
- ANYALPHA function 225
- ANYDIGIT function 225
- ANYPUNCT function 225
- ANYSPACE function 225
- APPEND procedure 478
- arithmetic operators 19–20
- array reference 245
- ARRAY statement
 - asterisk (*) in 247–248
 - changing array bounds 250–251
 - converting character values to lowercase 248–249
 - creating variables 249–250
 - missing character values in 247–248
 - missing numeric values in 245–246
 - table lookups 254–255
 - temporary arrays 251–252
- arrays
 - CALL routines and 246
 - changing bounds 250–251
 - converting character values to lowercase 248–249
 - creating variables 249–250
 - defined 244
 - loading initial values from raw data 253
 - missing character values in 247–248
 - missing numeric values in 244–246
 - multidimensional 254–257
 - table lookup and 254–257
 - temporary 251–257
- ASCII coding method 35, 230
- assignment statements 19–20, 23
 - defined 19
 - RETAIN statement and 473
- asterisk (*)
 - as wildcard 338, 446, 538
 - associating formats 374
 - in ARRAY statement 247–248
 - in assignment statements 19
 - in comment statements 19
 - TABLE statement (TABULATE) and 368
 - two-way tables and 356
- at sign (@)
 - absolute column pointer 456–457
 - column pointers and 40
 - double trailing (@@) 197, 454–455
 - format catalog and 465
 - informats and 488
 - INPUT statement and 197
 - single trailing 130, 451–454
- automatic macro variables 523
- AUTONAME option, OUTPUT statement (MEANS) 329–330, 337–338
- AXIS statement, GCHART procedure 413, 421
- ORDER= option 421

B

bar charts
 adding variables to 423–425
 creating for continuous variables 416–418
 producing 413–415
 representing means 422–423
 representing sums 420–422
 with values representing categories
 418–420

BEFORE option, RBREAK statement
 (REPORT) 303

BEGINNING alignment, INTNX function 156

BETWEEN AND operator 113

blanks
 concatenation operator and 366
 converting multiple 214–215
 dividing strings into words 230
 IN operator and 267
 missing character values and 192
 raw data separated by 30–31
 removing trailing/leading 217–218,
 233–234
 searching for 225
 TABULATE operators and 366, 368

BODY= keyword 400

Boolean operators 107, 109–112

BREAK statement, REPORT procedure
 303–306
 SKIP option 305
 SUMMARIZE option 305
 SUPPRESS option 306

Burlew, Michelle 522

BY groups
 computing differences between first/last
 observations 514–516
 counters and 508–511

BY statement
 adding subtotals/totals to listings 274–276
 CLASS statement and 327
 easier to read listings 277–278
 MEANS procedure and 323–324, 327,
 330–331
 merging data sets 171–173, 181–182

merging data sets with different data types
 179–181
 merging data sets with different names
 177–178
 omitting in merges 172–173
 outputting summary data sets 330–332
 SET statement and 167–168, 507–508

BY SUBJECT statement 498

C

CALL MISSING routine 193, 246, 497

CALL routines 193
 arrays and 246
 restructuring data sets with DATA step 497

CALL SYMPUT routine 531

CARDS statement 36

Carpenter, Art 522

Cartesian product 539–542, 551–552

cases
 changing 213–214
 SAS and 7
 searching for 225–226

CAT function 215–217

CATALOG procedure 465

Cates, Randall 438

CATS function 215–217, 487, 489

CATX function 215–216

CENTER option, DEFINE statement (REPORT)
 295

character classes
 ANY functions and 225–226
 defined 219
 NOT functions and 226–227

character functions
 ANY functions 225–226
 changing character case 213–214
 comparing strings 232–234
 concatenating strings 215–217
 data cleaning with 227–228
 determining value lengths 212–213
 dividing strings into words 230–232
 extracting parts of strings 228–230
 fuzzy matching with 234–235

- NOT functions 226–227
- removing characters from strings 214–215, 218–220
- removing trailing/leading blanks 217–218
- searching for character classes 225–226
- searching for characters 220–223
- searching for words in strings 223–225
- substituting characters/words 235–238
- `_CHARACTER_` keyword 247–249
- character values
 - changing case of 213–214
 - character-to-numeric conversions 180, 201–202, 229, 256, 468–469
 - comparing 232–234
 - converting to lowercase 248–249
 - determining length of 212–213
 - fuzzy matching for 234–235
 - IN operator and 267
 - missing values in 192–193
 - numeric-to-character conversions 202
 - PUT function and 202
 - reading in one step 467–470
 - removing from strings 214–215, 218–220
 - removing trailing/leading blanks 217–218
 - replacing missing values for arrays 247–248
 - searching for 220–223
 - setting as missing 193
 - substituting 235–238
- character variables
 - categories and 365
 - COMPUTE blocks and 308–309
 - computing frequencies of 342
 - defined 8
 - detail reports for 292
 - DO loops and 130–131
 - dollar sign (\$) and 13
 - extracting parts of strings 228–230
 - formats with 74
 - INPUT function and 201
 - logical comparison operators and 107
 - replacing missing values for arrays 247–248
 - character-to-numeric conversions 229, 468–469
 - CHART procedure 412–413
 - CHARTYPE option, MEANS procedure 334–337
 - CLASS statement
 - BY statement and 327
 - complex tables 377–378
 - MEANS procedure and 324–325, 327, 333–337
 - missing values in TABULATE procedure 385–389
 - MLF option 483
 - outputting summary data sets 331–333
 - PRELOADFMT option 484
 - TABULATE procedure and 365
 - class variables
 - analysis variables and 372–373
 - applying formats to 325–326
 - computing percentages on 384
 - counting number of visits and 511–512
 - formats and 462–463
 - missing values and 386–388
 - multiple 333–337
 - NWAY option, MEANS procedure and 511
 - PCTN statistic and 379–380
 - TABULATE procedure and 365
 - CLM statistic 321
 - CNTLIN= option, FORMAT procedure 471–476, 479, 487
 - CNTLOUT= option, FORMAT procedure 477
 - colon (:)
 - as delimiter 35
 - as modifier 233
 - as wildcard 202, 337
 - informat and 44, 456
 - logical comparison operators and 107
 - color, setting 413
 - COLPCTN keyword 382–383
 - COLPCTSUM keyword 384

- column headings
 - labeling 273–274
 - modifying labels for ACROSS variable 311
 - renaming with SQL procedure 540–541
- column indices 254
- column input 37–39
- column pointers 40, 456–457
- COLUMN statement, REPORT procedure
 - adding 291
 - changing order of variables in 297–298
 - computing character variables 309
 - computing new variables 308
 - controlling order of variables 300–301
 - creating ACROSS variable 310
 - displaying statistics with ACROSS variable 312
 - grouping variables and 296–297
 - ordering reports with nonprinting variables 307
- columns
 - computing percentages 379–380, 382–385
 - crosstab tables and 356–357
 - displaying percentages in 381–382
 - fixed 37–43
 - TABLE statement (TABULATE) and 367
 - variables and 18, 31, 536
 - wrapping lines of text 294–296
- comma (,)
 - changing values appearances 265–266
 - column input and 37
 - comma informat 180
 - formatting bar charts 418
 - in CSV files 33
 - in multidimensional arrays 256
 - IN operator and 267
 - in TABLE statement (TABULATE) 367
- comma informat 180
- comma11. informat 180
- comment statements 19–21
- COMPARE function 232–234
- COMPBL function 214–215
- compile stage 22–23
- COMPRESS function
 - removing characters from strings 214, 218–220
 - removing dashes with 180
 - searching for characters 221–222
- COMPUTE blocks
 - computing character variables 308–309
 - creating 308
 - selecting variables for reports 291
- COMPUTE statement, REPORT procedure 308–309
- COMPUTED option, DEFINE statement (REPORT) 308
- concatenating
 - data sets 165, 168, 546–549
 - strings 215–217, 366
- concatenation operator 215–217, 366
- conditional processing
 - See also* IF statement
 - See also* WHERE statement
 - Boolean operators 107, 109–112
 - combining detail/summary data 168–169
 - DO UNTIL statement 131–134, 448
 - DO WHILE statement 131–135
 - ELSE IF statement 102–105
 - IN operator 107
 - reading data conditionally 451–453
 - restructuring data sets with DATA step 496
 - SELECT statement 108–109
 - subsetting IF statement 105–107
 - substituting for missing date values 151–152
- CONSTANT function 198–199
- constants
 - computing 198–199
 - date 147–148
 - hexadecimal 35
- CONTAINS operator 113–114
- CONTENTS= keyword 400
- CONTENTS procedure
 - _ALL_ keyword 59
 - conversion process and 98
 - documenting data sets with 80–81

- examining data sets with 56–58
 - listing data sets with 59
 - NODS option 59
 - VARNUM option 58, 149
 - CONTINUE statement 135–136
 - continuous variables
 - creating bar charts for 416–418
 - with values representing categories 418–420
 - converting
 - character values to lowercase 248–249
 - characters to numbers 180, 201–202, 229, 256, 468–469
 - data sets into CSV files 96–98
 - data sets into spreadsheets 93–95
 - Fahrenheit to Celsius 250
 - missing numeric values 244–246
 - multiple blanks 214–215
 - numbers to characters 202
 - spreadsheets into CSV files 87–88
 - spreadsheets with Import Wizard 88–92
 - with XLS engines 95–96
 - Corel WordPerfect 402
 - counters
 - arrays and 246
 - BY groups and 508
 - FREQ procedure and 509–511
 - in DATA step 253
 - setting 508
 - sum statement and 120, 124
 - CREATE clause (SQL) 538
 - crosstab tables 356–358
 - CSV files
 - converting data sets into 96–98
 - converting spreadsheets into 87–88
 - embedded delimiters in 46
 - informats and 44
 - reading data values 33
 - CTRL+C key combination 134
 - curly brackets { } 245, 254
 - current date 148–149
 - customized reports
 - applying ORDER usage to variables 300–301
 - changing order of variables in 297–298
 - changing row order in 299–300
 - comparing detail/summary reports 291–293
 - COMPUTE blocks in 308–309
 - computing new variables for 307–308
 - creating ACROSS variable 310
 - displaying statistics with ACROSS variable 311–313
 - FLOW option, REPORT procedure 294–296
 - grouping variables 296–297
 - modifying labels for ACROSS variable 311
 - multi-column 301–302
 - ordering with nonprinting variables 306–307
 - producing breaks in 303–306
 - producing summary reports 293–294
 - REPORT procedure and 288–290
 - selecting variables for 291
- D**
- dash (-) 180
 - data cleaning
 - NOT functions for 226–227
 - VERIFY function 227–228
 - with character functions 227–228
 - DATA_NULL_ reporting 68, 444
 - DATA= option, SURVEYSELECT procedure 200
 - data portion (data sets)
 - defined 56
 - viewing 63–64
 - viewing with SAS VIEWTABLE window 64–65
 - data sets 8
 - See also* merging data sets
 - See also* permanent data sets
 - See also* summary data sets
 - accessing with user-defined formats 82

- data sets (*continued*)
 - adding observations to 164–167
 - combining detail/summary data 168–170
 - concatenating 165, 168, 546–549
 - controlling observations in 173–175
 - converting spreadsheets to 88–92
 - converting via ODS 96–98
 - creating formats 471–476
 - creating spreadsheets from 93–95
 - descriptor portion of 22, 56–58, 60–63, 73
 - documenting 80
 - interleaving 167–168
 - JOIN option, SYMBOL statement 429
 - naming conventions 7
 - naming variables in output 329–330
 - output 329–330, 408–409
 - permanent attributes for 80–81
 - restructuring using DATA step 494–497
 - restructuring using TRANSPOSE procedure 497–500
 - SAS processing 24
 - sending output to 407–409
 - subsetting 112, 162–164
 - tables and 536
 - updating master files 183–184
 - virtual 474
 - WHERE statement and 112
- DATA step
 - combining detail/summary data 169
 - counters in 253, 508
 - creating labels in 72–73
 - creating summary data sets 336–337
 - data sets as input to 65–66
 - defined 6
 - end of file and 176–177
 - FORMAT statement in 43, 79–80
 - INPUTN function in 485–490
 - LABEL statement in 73, 79–80
 - labeling column headings 273
 - %LET statement and 524
 - nested formats in 480–481
 - _NULL_ keyword and 67–68
 - restructuring data sets using 494–497
 - SAS processing 22–24
 - semi-colon (;) and 36
 - SET statement and 177
 - SQL procedure and 536, 549
 - subsetting data steps 163–164
 - transferring values between 530–532
- data structures, reading 456–457
- data summaries
 - See* summarizing data
- data types 8, 179–181
- data view 474
- DATALINES statement 36–37, 448
- date constant 147–148
- DATE function 149
- date interval functions 152–157
- date9. format 43, 145, 523
- dates
 - automatic macro variables and 523
 - computing current 148–149
 - computing years between 146–147
 - creating from day values 150–151
 - creating from month values 150–151
 - creating from year values 150–151
 - extracting day of month from 149–150
 - extracting day of week from 149–150
 - extracting year from 149–150
 - INPUT function and 201
 - interval functions for 152–157
 - reading values from raw data 143–145
 - storing 142
 - substituting missing values for 151–152
- day of month
 - extracting 149–150
 - substituting for missing values 151–152
- day of week 149–150, 419
- debugging 68
- DEFINE statement, REPORT procedure
 - ACROSS option 310, 311–313
 - ANALYSIS option 292–295, 312
 - CENTER option 295
 - COMPUTED option 308
 - creating ACROSS variable 310
 - DISPLAY option 292–293, 295

- displaying statistics with ACROSS variable
 - 311–313
- FLOW option 294–296
- GROUP option 293–294, 296–297, 303–305
- LEFT option 295
- MEAN option 293–294, 312
- modifying column label for ACROSS variable 311
- NOPRINT option 307–308
- ORDER= option 299–301, 303–305
- ordering reports with nonprinting variables 307
- RIGHT option 295
- DELETE statement 120, 454
- DELIMITER= option, INFILE statement 35
- delimiters
 - blanks as 30–32
 - commas as 33
 - defined 23–24
 - dividing strings into words 230
 - DLM= option for 34–35
 - embedded in list input 46
- DESCENDING option
 - ORDER option, DEFINE statement (REPORT) 300–301
 - SORT procedure 270–271
- descriptive statistics
 - outputting with MEANS procedure 328–329
 - TABULATE procedure and 370–372
- descriptive statistics functions 194–196
- descriptor portion (data sets) 22
 - examining 56–58
 - labels in 73
 - viewing with SAS Explorer 60–63
- detail reports 291–293
- DIF function 204, 207, 513
- digits, searching for 225
- DIM function 248
- DISCRETE option, VBAR statement (GCHART) 419–420
- Display Manager 9, 406
- DISPLAY option, DEFINE statement (REPORT) 292–293, 295
- displaying data 262–263
 - adding number of observations to listings 279
 - adding subtotals/totals to listings 274–277
 - adding titles/footnotes to listings 268–270
 - changing listing appearance 263–265
 - changing listing order 270–272
 - changing values appearances 265–266
 - controlling observation appearance in listings 266–267
 - double-spacing listings 280
 - easier to read listings 277–278
 - labeling column headings 273–274
 - listing specified number of observations 281–283
 - sorting by multiple variables 272–273
- division in assignment statements 19–20
- DLM= option, INFILE statement 34–35, 37
- DO statement
 - arrays in 246
 - converting character values to lowercase 249
 - DO groups and 119
 - iterative looping 125–129
 - iterative processing and 118–120
 - multidimensional arrays and 256
 - other forms 129–131
- DO UNTIL statement 131–134, 448
- DO WHILE statement 131–135
- documenting data sets 80
- DOL option, RBREAK statement (REPORT) 303
- dollar sign (\$)
 - changing values appearances 265–266
 - column input and 37
 - formats and 74
 - informat and 180, 465
 - variable names and 13, 31
- dollar11.2 format 43, 75
- DONUT statement 414
- DOUBLE option, PRINT procedure 280

- double trailing at sign (@@) 197, 454–455
 - double-spacing listings 280
 - DROP= data set option
 - counting number of visits and 510, 512
 - DROP statement and 163
 - variable selection and 337
 - DROP= option, TRANSPOSE procedure 499
 - DROP statement
 - DROP= data set option and 163
 - dropping variables from data sets 337
 - retained variables and 516
 - shortening 202
 - DSD= option, INFILE statement
 - CSV files and 33, 88
 - DATALINES statement and 37
 - DLM= option and 35
 - DUL option, RBREAK statement (REPORT) 303
- E**
- e (mathematical constant) 198–199
 - EBCDIC coding method 35, 230
 - ELSE IF statement 102–105
 - embedded delimiters 46
 - END alignment, INTNX function 156
 - END= data set option 475, 478
 - end of file
 - DATA step and 176–177
 - detecting 443–445
 - end of line 438–440
 - END= option
 - INFILE statement 443–446
 - SET statement 445
 - END statement
 - DO groups and 119
 - iterative DO loop and 126, 134
 - LEAVE statement and 135
 - ENDCOMP statement, REPORT procedure 308–309
 - engines
 - conversion process and 54
 - reading spreadsheets with 95–96
 - Enterprise Guide 9
 - EQ operator 103
 - equal sign (=)
 - formats and 74, 78
 - in labels 72
 - WHERE statement operator and 113
 - equations, graphing 128–129
 - Excel spreadsheets
 - converting into CSV files 87–88
 - converting with Import Wizard 88–92
 - converting with ODS 96–98
 - creating from data sets 93–95
 - reading with engines 95–96
 - EXCEPT operator 546
 - EXCLUDE statement, FORMAT procedure 84
 - execution stage 22–24
 - EXP function 197–198
 - Explorer
 - conversion process and 98
 - documenting data sets with 80
 - viewing data sets with 60–63
 - exponentiation in assignment statements 19–20
 - EXPORT statement 95
 - Export Wizard 93–95
 - external files
 - alternative methods for 34
 - PUT statement and 202
 - reading 447–448
 - reading long 443
- F**
- Fahrenheit-to-Celsius conversion 250
 - FancyPrinter style 401
 - FILE PRINT statement 444
 - FILENAME statement
 - reading external files 447–448
 - reading from multiple files 446–447
 - sending output to HTML files 399
 - specifying external files 34
 - filerefs 34, 448
 - FILEVAR option, INFILE statement 448
 - FIND function 221–224
 - FINDC function 223
 - FINDW function 223–225

- FIRSTOBS= data set option 92, 282
- FIRSTOBS= option, INFILE statement 445–446
- fixed columns
 - column input 37–39
 - formatted input 39–43
- FLOW option, DEFINE statement (REPORT) 294–296
- FMTLIB option, FORMAT procedure
 - format definitions and 82, 84
 - listing formats 473, 479
 - SELECT statement and 488
 - viewing catalog entries 465
- FMTSEARCH= system option 80, 82
- folders 59
- fonts, setting 413
- FOOTNOTE statement
 - displaying data with 268, 270
 - RESET=all graphics option and 412
- footnotes 268–270
- format catalog 465
- format definitions 82–84
- format library 79
- FORMAT= option, TABULATE procedure 366
- FORMAT procedure
 - CNTLIN= option 471–476, 479, 487
 - CNTLOUT= option 477
 - creating numeric informats 487
 - enhancing output with 73–74
 - EXCLUDE statement 84
 - FMTLIB option 82, 84, 465, 473, 479, 488
 - INVALUE statement 465–466, 469
 - LIBRARY= option 79
 - SELECT statement 84, 488
 - storing formats 79
 - user-defined formats 380, 464
 - VALUE statement 74, 78, 482–485
- FORMAT statement
 - applying formats to class variables 325–326
 - associating formats with 41–42, 73
 - changing values appearances 265–266
 - for bar charts 418
 - formatting date values 144
 - in DATA step 43, 79–80
 - labeling output 346–347
 - TABLES statement and 77
- formats 41–43
 - applying to class variables 325–326
 - associating with asterisk (*) 374
 - creating 73, 471–476
 - DATA_NULL_ reporting and 68
 - enhancing output with 73–76
 - for date values 144–145
 - formats within 479–481
 - in DATA step 43
 - labeling output with 346–347
 - listing for variables 80
 - maintaining 477–479
 - multi-label 482–485
 - problems when grouping values 349–350
 - PUT function and 463–464
 - recoding variables with 462–463
 - regrouping values using 76–77
 - specifying ranges 78–79
 - storing 79
 - table lookup and 470–471
 - to group values 347–348
 - updating 477–479
 - user-defined 79–82, 380, 464
- formatted input 39–43
- forward slash (/)
 - as relative line pointer 450
 - in assignment statements 19
 - statement options and 330
- four-digit years 145
- FRAME= keyword 400
- FREQ procedure 13, 342–344
 - See also* TABLES statement, FREQ procedure
 - changing order of values in 353–356
 - counting number of visits with 509–511
 - displaying missing values 351–352
 - formats and 74, 77, 462
 - grouping values through formats 347–348

- FREQ procedure (*continued*)
 - labeling output with formats 346–347
 - listing observations per quarter 154
 - multiple two-way tables 358
 - NOPRINT option 509
 - ORDER= option 353–356
 - output example 402–403
 - problems when grouping values 349–350
 - producing three-way tables 358–360
 - producing two-way tables 356–357
 - sample SAS program 16
 - selecting variables for 345–346
 - FREQ variable 331, 337
 - frequencies
 - See FREQ procedure
 - FROM clause (SQL) 537, 539–542
 - full joins 543–545
 - fuzzy matching/merge 234–235, 541, 551–552
- G**
- GCHART procedure
 - adding variables to charts 423–425
 - AXIS statement 413, 421
 - bar charts representing means 422–423
 - bar charts representing sums 420–422
 - charts with values representing categories 418–420
 - creating bar charts for continuous variables 416–418
 - creating pie charts 415–416
 - PIE statement 414–415
 - producing bar charts 413–415
 - VBAR statement 414, 417–424
 - GE operator 103
 - GLM procedure 463
 - global statements 6, 13
 - GOPTIONS statement 413–414
 - VSIZE= option 414
 - GPLOT procedure
 - See also SYMBOL statement, GPLOT procedure
 - example of 129, 154
 - PLOT statement 426
 - producing scatter plots 425–427
 - grand mean 332
 - graphics 412–413
 - adding variables to charts 423–425
 - bar charts representing means 422–423
 - bar charts representing sums 420–422
 - charts with values representing categories 418–420
 - connecting points 427–429
 - connecting points with smooth line 429–430
 - creating pie charts 415–416
 - creating pie charts for continuous variables 416–418
 - producing bar charts 413–415
 - producing scatter plots 425–427
 - graphing equations 128–129
 - Gregorian calendar 41
 - GROUP option, DEFINE statement (REPORT) 293–294, 296–297, 303–305
 - GROUP= option, VBAR statement (GCHART) 423
 - grouping
 - values through formats 347–348
 - values with FREQ procedure 349–350
 - variables 296–297
 - GT operator 103
- H**
- Haworth, Lauren 382
 - HAXIS option, PLOT statement (GPLOT) 426
 - HBAR statement 414
 - HBAR3D statement 414
 - HEADING= option, PRINT procedure 282
 - HEADLINE option, REPORT procedure 297
 - hexadecimal constants 35
 - HTML files
 - creating table of contents 400–401
 - selecting different styles 401–402
 - sending output to 398–399
 - hyphen (-) 180

I

- ICD-9 codes 472–477
- ID statement
 - controlling listing appearance 264–265
 - easier to read listings 277–278
 - variables in 75
- IF statement
 - See also* subsetting IF statement
 - arrays and 246
 - computing differences between observations 513
 - conditional processing 102–105
 - DO groups and 119–120
 - example 67
 - in procedures 112
 - missing character values in arrays 248
 - MISSING function 104
 - restructuring data sets with DATA step 496
 - substituting for missing date values 151–152
- IMPORT procedure 91
- Import Wizard 88–92
- IN= data set option
 - checking missing values 175–176
 - controlling observations with 173–175
- IN operator
 - conditional processing 107
 - controlling observation appearance in listings 267
 - listed 103
- INDEX function 222
- index variables 128–130
- INDEXW function 223
- INFILE statement
 - DELIMITER= option 35
 - DLM= option 34–35, 37
 - DSD= option 33, 35, 37, 88
 - END= option 443–446
 - filerefs in 34
 - FILEVAR option 448
 - FIRSTOBS= option 445–446
 - LRECL option 443
 - MISSOVER option 443
 - OBS= option 445–446
 - options in DATALINES statement 37
 - PAD option 143, 442–443
 - PUT statement and 68
 - reading external filenames 447–448
 - reading long external files 443
 - reading raw data with 12, 31
 - SAS processing 22
 - TRUNCOVER option 143, 443
- infinite loops 134–135
- informat lists 455–456
- informat modifiers 44, 46
- INFORMAT statement 45
- informats 40–41
 - at sign (@) and 488
 - colon (:) and 44, 456
 - creating numeric 488–489
 - defined 40
 - INFORMAT statement 45
 - INPUT function and 201–202
 - INPUTN function and 485–490
 - reading data in one step 467–470
 - reading date values from raw data 143
 - table lookup and 470–471
 - user-defined 464–467
 - variable lists and 455–456
 - with list input 43–44
- inner joins 543–545
- input buffer 22
- INPUT function 201–202
 - character-to-numeric conversion 180, 201–202, 229, 256, 468–469
 - nested formats and 481
 - table lookups and 471, 486, 488–489
 - user-defined informats and 464, 466
- INPUT statement
 - ampersand (&) modifier in 46
 - at sign (@) in 197
 - CSV files and 88
 - informat lists and 455–456
 - INFORMAT statement and 45
 - informats and 43–44
 - INPUT function and 202

- missing values at end of line 438–440
- multiple lines of data for observations 448–450
- multiple observations from line of input 454–455
- reading data conditionally 451–453
- reading raw data with 12, 31
- reading short data lines 440–443
- relative column pointers 456–457
- SAS processing 22, 24
- single trailing at sign (@) 130
- trailing at sign (@) and 454
- variable lists and 455–456
- INPUTN function 485–490
- INT function 190–191
- INTCK function 152–155
- interleaving data sets 167–168
- INTERPOL= option, SYMBOL statement 427–429
- INTERSECTION operator 546
- INTNX function 152–153, 155–157
- INVALUE statement, FORMAT procedure 465–466, 469
 - JUST option 466
 - UPCASE option 466, 469
- IS MISSING operator 113
- IS NULL operator 113
- iterative DO loop 125–129
 - other forms 129–131
- iterative processing
 - See* looping
- J**
- JOIN option, SYMBOL statement 428–429
- JOURNAL style 402
- JUST option, INVALUE statement (FORMAT) 466
- K**
- KEEP= data set option 163, 337, 510
- KEEP statement 496
- KEYLABEL statement, TABULATE procedure 375–376, 380, 383
- keywords, renaming 375
- L**
- LABEL option, PRINT procedure 273, 279
- LABEL statement 72
 - adding number of observations to listings 279
 - in DATA step 73, 79–80
 - labeling column headings 273–274
- labels
 - adding to variables 71–73
 - defining format 479
 - for column headings 273–274
 - formats for 346–347
 - listing for variables 80
 - modifying for ACROSS variable 311
 - multi-label formats 482–485
- Lafler, Kirk 536
- LAG function 204–207, 512–515
- LARGEST function 195
- LE operator 103
- leading blanks, removing 217–218
- LEAVE statement 135–136, 496
- LEFT function 217–218
- left joins 543–545
- LEFT option, DEFINE statement (REPORT) 295
- LENGTH function 212–213
- LENGTH statement
 - CONSTANT function and 199
 - dividing strings into words 231
 - extracting parts of strings 229
 - index variables and 130
 - maintaining formats 478
 - SET statement and 167
- LENGTHC function 213
- LENGTHN function 212–213
- less than sign (<) 78
- %LET statement 524–525
- LIBNAME statement 54–55, 58
- libraries 59
- LIBRARY= option, FORMAT procedure 79

- librefs
 - defined 54–55
 - macro variables specifying 529–530
 - storing formats 79
 - LIKE operator 113–114
 - LINE= option, SYMBOL statement 429
 - line pointers 450
 - LINESIZE= system option 282
 - list input
 - blanks in 30–31
 - defined 23
 - INFORMAT statement with 45
 - informats with 43–44
 - missing values at end of line 438–440
 - specifying missing values 32
 - with embedded delimiters 46
 - LISTING destination 408
 - listings
 - See also* reports
 - adding number of observations to 279
 - adding subtotals/totals to 274–277
 - changing appearance of 263–265
 - changing order of 270–272
 - CONTENTS procedure and 59
 - controlling observation appearance in 266–267
 - double-spacing 280
 - easier to read 277–278
 - formats and 74
 - OBS= option 281–283
 - ODS statement and 399
 - PRINT procedure and 63–64
 - LOG function 197–198
 - Log window 15–16
 - LOG10 function 197–198
 - logical comparison operators
 - Boolean logic 107, 109–112
 - conditional processing and 107
 - listed 103
 - longitudinal data 506
 - looping
 - arrays and 246
 - CONTINUE statement 135–136
 - converting character values to lowercase 249
 - DO groups and 118–120
 - DO UNTIL statement 131–134
 - DO WHILE statement 131–135
 - infinite 134–135
 - iterative DO loop 125–131
 - LEAVE statement 135–136
 - multidimensional arrays and 256
 - restructuring data sets with DATA step 496
 - sum statement and 120–125
 - LOWCASE function 214, 235
 - lowercase
 - converting character values to 248–249
 - LOWCASE function 214, 235
 - LRECL option, INFILE statement 443
 - LT operator 103
- M**
- %MACRO statement 525
 - macro variables
 - as prefixes 529–530
 - assigning values with %LET statement 524–525
 - automatic 523
 - built-in 523
 - defined 522
 - tokens and 527–529
 - transferring values between DATA steps 530–532
 - macros 525–527
 - many-to-many merge 182
 - master files, updating 183–184
 - mathematical functions 197–199
 - MAX function 195
 - MAX statistic 321
 - MAXDEC statistic 321
 - MAXIS= option, VBAR statement (GCHART) 421
 - MDY function 150–152
 - MEAN function 194, 549–550
 - MEAN option, DEFINE statement (REPORT) 293–294, 312

- MEAN statistic 321, 327–328, 378
- means, bar charts representing 422–423
- MEANS procedure 14, 320–322
 - applying formats to class variables 325–326
 - BY statement and 323–324, 327, 330–331
 - CHARTYPE option 334–337
 - CLASS statement with 324–325, 327, 333–337
 - combining detail/summary data 169
 - counting number of visits with 509, 511–512
 - creating summary data sets 327–328
 - formats and 74, 462–463
 - labels example 72–73
 - macro variables transferring values 530–532
 - multilabel formats 482–483
 - multiple class variables with 333–337
 - NOPRINT option 327–328, 511–512
 - NWAY option 332–333, 336, 511
 - OUTPUT statement 327, 329–330, 337–338
 - outputting descriptive statistics with 328–329
 - outputting summary data sets 330–333
 - sample SAS program 16
 - selecting statistics for variables 337–338
 - sending output to HTML files 398
 - SQL procedure and 549
 - statistic options listed 321
 - VAR statement and 322, 329–330
- MEDIAN statistic 321
- %MEND statement 525
- MERGE statement 171, 182, 510
- MERGENOBY system option 173
- merging data sets 170–172
 - controlling observations 173–175
 - many-to-many 182
 - omitting BY statement 172–173
 - one-to-many 181
 - one-to-one 181
 - with different data types 179–181
 - with different names 177–178
- merging tables 539–545
- METHOD= option, SURVEYSELECT procedure 200
- Microsoft Office Word 402
- MIDDLE alignment, INTNX function 156
- MIDPOINTS= option, VBAR statement (GCHART) 417–418
- MIN function 195–196
- MIN statistic 321
- MISSING function
 - IF statement and 104
 - numeric functions and 192–193
 - substituting for missing date values 152
 - testing for missing values 496
 - true value for 120
- MISSING option
 - TABLES statement (FREQ) 351–352
 - TABULATE procedure 387–388
- MISSING routine 193, 246, 497
- missing values
 - adding observations to data sets 166
 - at end of line 438–440
 - checking with IN= data set option 175–176
 - conditional processing and 103
 - DATA step and 66
 - FREQ procedure and 351–352
 - grouping problem with 349–350
 - in numeric functions 192–193
 - on class variables 386
 - printing 485
 - replacing for character variables 247–248
 - replacing for numeric variables 244–246
 - setting 193
 - specifying with list input 32
 - substituting for dates 151–152
 - sum statement and 123
 - table lookups and 471
 - TABULATE procedure and 385–389
 - testing for 496
- MISSOVER option, INFILE statement 443
- MISSPRINT= option, TABLES statement (FREQ) 388

- MISSTEXT= option, TABLE statement (TABULATE) 389, 485
 - MLF option, CLASS statement 483
 - mmddy10. format 145, 480
 - mmddy10. informat 40–42, 479
 - modifiers
 - COMPARE function and 233
 - COMPRESS function and 219–220
 - defined 219
 - informat 44, 46
 - MONTH function 149
 - months
 - creating dates from 150–151
 - date interval functions 152–157
 - extracting from dates 149–150
 - MPRINT system option 525
 - multi-column reports 301–302
 - multidimensional arrays 254–257
 - MULTILABEL option, VALUE statement (FORMAT) 482–485
 - multi-level sorts 272–273
 - multiplication in assignment statements 19–20
- N**
- N function 194–195
 - N= option, PRINT procedure 279
 - N statistic 321, 375, 378
 - NA value 247–248
 - names 7–8
 - naming conventions
 - data sets 7
 - librefs 55
 - variables 7
 - NE operator 103
 - negation in assignment statements 19–20
 - nested formats 479–481
 - nesting operator 368
 - NMISS function 195
 - NMISS statistic 321, 375
 - NOCENTER system option 16, 263
 - NOCOL option, TABLES statement (FREQ) 359
 - NOCUM option, TABLES statement (FREQ) 345
 - NODS option, CONTENTS procedure 59
 - NOHEADING option, PIE statement (GCHART) 415
 - NOOBS option, PRINT procedure 97, 265
 - NOPERCENT option, TABLES statement (FREQ) 346, 359
 - NOPRINT option
 - DEFINE statement (REPORT) 307–308
 - FREQ procedure 509
 - MEANS procedure 327–328, 511–512
 - procedures and 408
 - NOROW option, TABLES statement (FREQ) 359
 - NOSEPS option, TABULATE procedure 376, 381
 - NOT functions 226–227
 - NOT operator 109–111
 - NOTALNUM function 227
 - NOTALPHA function 227
 - NOTDIGIT function 226–227
 - NOWD option, REPORT procedure 289–290
 - _NULL_ keyword 67–68
 - numeric functions
 - computing constants with 198–199
 - computing sums with 196–197
 - descriptive statistics functions 194–196
 - generating random numbers 199–201
 - mathematical functions 197–198
 - missing values in 192–193
 - return values from observations 204–207
 - rounding numeric values 190–191
 - setting missing values 193
 - special functions 201–203
 - truncating numeric values 190–191
 - numeric values
 - character-to-numeric conversions 180, 201–202, 229, 256, 468–469
 - IN operator and 267
 - missing values in 192–193
 - numeric-to-character conversions 202
 - reading in one step 467–470

- numeric values (*continued*)
 - replacing missing values for arrays 244–246
 - rounding 190–191
 - truncating 190–191
- numeric variables
 - computing frequencies of 342
 - computing percentages on 384–385
 - computing statistics on 14, 321
 - defined 8
 - informats and 467–470
 - logical comparison operators and 107
 - replacing missing values for arrays 244–246
 - summary reports for 292
- NWAY option, MEANS procedure 332–333, 336, 511
- O**
- OBS= data set option 92, 281–283
- OBS= option, INFILE statement 445–446
- observations
 - adding to data sets 164–167
 - adding to listings 279
 - checking missing values for 175–176
 - combining detail/summary data 168–170
 - computing differences between 512–514
 - computing differences between first/last 514–516
 - computing sums within 196–197
 - controlling appearance in listings 266–267
 - controlling in merged data sets 173–175
 - counting number of visits 509–512
 - detail reports about 291
 - functions returning values from 204–207
 - identifying first/last in groups 506–509
 - listing per quarter 154
 - listing specified number of 281–283
 - multiple 454–455
 - reading multiple lines from 448–450
 - restructuring data sets using DATA step 494–497
 - restructuring data sets using TRANSPOSE procedure 497–500
 - retained variables and 515–517
 - table rows and 18, 31, 536
- ODS (Output Delivery System)
 - choosing destinations 402–403
 - converting data sets into spreadsheets 96–98
 - creating table of contents 400–401
 - procedures and 397–398
 - selecting different HTML styles 401–402
 - selecting/excluding output 403–407
 - sending output to data sets 407–409
 - sending output to HTML files 398–399
- ODS CLOSE statement 97
- ODS CSV statement 97
- ODS EXCLUDE statement 403, 406–407
- ODS HTML CLOSE statement 399
- ODS HTML FILE statement 399
- ODS HTML statement 400
- ODS OUTPUT statement 408
- ODS SELECT statement 403–407, 409
 - PERSIST option 407
- ODS statement 399
- ODS TRACE statement 404–406, 408
- OL option, RBREAK statement (REPORT) 303
- ON clause (SQL) 543–545
- one-to-many merge 181
- one-to-one merge 181
- operators
 - arithmetic 19–20
 - asterisk (*) as 368
 - Boolean 107, 109–112
 - comma as 367
 - concatenation 215–217, 366
 - for TABULATE procedure 366–368
 - in WHERE statement 113–114
 - logical comparison 103, 107
 - UNION 546–549
- OR operator 107, 109–112
- ORDER clause (SQL) 551
- ORDER= option
 - AXIS statement (GCHART) 421

- DEFINE statement (REPORT) 299–301, 303–305
 - FREQ procedure 353–356
 - OTHER keyword 349–350, 471, 475
 - OTHERWISE statement 108–109
 - OUT= option
 - OUTPUT statement, MEANS procedure 327
 - procedures and 407
 - SORT procedure 271
 - SURVEYSELECT procedure 200
 - output
 - See also* ODS (Output Delivery System)
 - choosing destinations 402–403
 - for summary data sets 330–333
 - formats in 73–76
 - labeling with formats 346–347
 - missing values in TABULATE procedure 385–389
 - selecting/excluding portions of 403–407
 - sending to data sets 407–409
 - sending to HTML files 398–399
 - output data sets
 - creating simplified reports with 409
 - determining structure of 408
 - naming variables in 329–330
 - Output Delivery System
 - See* ODS
 - output objects 398, 404–406
 - OUTPUT statement
 - counting number of visits and 512
 - iterative DO loop 126, 128–129
 - SAS processing 24
 - subsetting data sets 164
 - OUTPUT statement, MEANS procedure 327, 329–330, 337–338
 - AUTONAME option 329–330, 337–338
 - OUT= option 327
 - Output window 15, 67–68
- P**
- PAD option, INFILE statement 143, 442–443
 - PAGEBY statement 276
 - PANELS= option, REPORT procedure 301–302
 - parentheses ()
 - ARRAY statement and 245
 - Boolean operators and 110
 - in assignment statements 20
 - logical comparison operators and 107
 - variable lists in 455
 - PATH statement 400
 - PATTERN statement 412–414, 425
 - PCTN statistic 379–380, 383
 - PCTSUM statistic 384
 - PDF output destination 402–403
 - PDV (Program Data Vector) 22–24
 - adding observations to data sets 166–167
 - combining detail/summary data 169
 - merging data sets with different names 178
 - missing character values in arrays 247
 - RETAIN statement and 516
 - subsetting data sets 163
 - PERCENT format 169, 531
 - percent sign (%) 114, 522
 - percentages
 - computing 379–380, 382–383
 - computing on numeric variables 384–385
 - in two-dimensional tables 381–382
 - period (.)
 - list input and 32
 - macro processor and 529–530
 - missing values and 192, 388–389
 - permanent data sets and 54–55
 - permanent data sets
 - as input to DATA step 65–66
 - examining with CONTENTS procedure 56–58
 - LIBNAME statement and 54–55
 - listing with CONTENTS procedure 59
 - _NULL_ keyword and 67–68
 - reason for creating 55
 - user-defined formats with 79–82
 - viewing with PRINT procedure 63–64
 - viewing with SAS Explorer 60–63

- permanent data sets (*continued*)
 - viewing with SAS VIEWTABLE window 64–65
- PERSIST option, ODS SELECT statement 407
- pi (mathematical constant) 198–199
- pie charts 415–416
- PIE statement, GCHART procedure 414–415
 - NOHEADING option 415
- PIE3D statement 414
- PLOT procedure 412
- PLOT statement, GPLOT procedure 426
 - HAXIS option 426
 - VAXIS option 426
- plus sign (+) 123, 456–457
- Prairie, Katherine 536
- PREFIX= option, TRANSPOSE procedure 500
- PRELOADFMT option, CLASS statement 484
- PRINT procedure
 - adding 31
 - adding number of observations to listings 279
 - adding subtotals/totals to listings 274–277
 - adding titles/footnotes to listings 268–270
 - changing listing appearance 263–265
 - changing listing order 270–272
 - changing values appearances 265–266
 - controlling observation appearance in listings 266–267
 - customized reports and 288
 - displaying data with 262–263
 - DOUBLE option 280
 - double-spacing listings 280
 - easier to read listings 277–278
 - FORMAT statement 41–42, 75, 265–266
 - HEADING= option 282
 - ID statement and 75
 - LABEL option 273, 279
 - labeling column headings 273–274
 - listing specified number of observations 281–283
 - N= option 279
 - NOOBS option 97, 265
 - output data sets and 408
 - REPORT procedure and 283, 288, 290
 - sending output to HTML files 398
 - SORT procedure and 299
 - sorting by multiple variables 272–273
 - viewing data sets with 63–64
 - WHERE statement and 336
- PRINTMISS option, TABLE statement (TABULATE) 484–485
- PRINTTO procedure 407
- PROC steps
 - creating labels in 72–73
 - defined 6
 - %LET statement and 524
 - SAS processing 24
- procedures
 - FORMAT statement and 43
 - IF statement in 112
 - NOPRINT option in 408
 - ODS and 397–398
 - OUT= option in 407
- Program Data Vector
 - See* PDV
- programs, SAS
 - See* SAS programs
- PROPCASE function 214–215
- punctuation
 - dividing strings into words 230
 - searching for 225
- PUT function 201–203
 - creating variables with 463–464
 - formats and 463–464
 - merging data sets 180
 - nested formats and 481
 - table lookups and 471
- PUT statement
 - controlling observations example 174
 - end of file and 444
 - in DATA step 67–68
 - PUT function and 202
- PUTC function 489
- PUTN function 489
- p*-values 407–409

Q

Q1 statistic 321
 Q3 statistic 321
 QRANGE statistic 321
 quarters, date interval functions 152–157
 queries
 Cartesian product 539
 demonstrating 537–538
 question mark (?) 247–248, 446
 QUIT statement 537
 quotation marks ("")
 in TITLE statement 57–58
 macro variables and 523
 missing character values and 192
 XLS engine and 96

R

random numbers, generating 199–201, 524
 RANUNI function 199–201
 raw data
 loading initial values from arrays 253
 reading 11–18
 reading column input 37–39
 reading date values from 143–145
 reading formatted input 39–43
 reading from multiple files 446
 reading from multiple files with
 FILENAME statement 447
 reading portion of 445–446
 reading short data lines 441–442
 separated by blanks 30–31
 separated by commas 33
 RBREAK statement, REPORT procedure
 303–306
 AFTER option 303
 BEFORE option 303
 DOL option 303
 DUL option 303
 OL option 303
 SUMMARIZE option 303
 UL option 303
 reading
 character data in one step 467–470

complex data structures 456–457
 data conditionally 451–453
 date values from raw data 143–145
 external files 447–448
 from multiple files 446
 from multiple files with FILENAME
 statement 447
 long external files 443
 multiple lines of data for observations
 448–450
 numeric data in one step 467–470
 portion of raw data file 445–446
 raw data 11–18
 raw data separated by blanks 30–31
 raw data separated by commas 33
 raw data with column input 37–39
 raw data with formatted input 39–43
 short data lines 440–443
 spreadsheets with engines 95–96
 relative column pointers 456–457
 relative line pointers 450
 RENAME= data set option
 counting number of visits and 510, 512
 renaming variables 177–178, 473
 SET statement and 202
 RENAME= option, TRANSPOSE procedure
 499
 RENAME statement 337
 REPORT procedure 289–290
 See also COLUMN statement, REPORT
 procedure
 See also DEFINE statement, REPORT
 procedure
 applying ORDER usage to variables
 300–301
 BREAK statement 303–306
 changing row order 299–300
 comparing detail/summary reports
 291–293
 COMPUTE blocks in 308–309
 COMPUTE statement 308–309
 computing new variables 307–308
 creating ACROSS variable 310

- REPORT procedure (*continued*)
- displaying statistics with ACROSS variable 311–313
 - ENDCOMP statement 308–309
 - grouping variables 296–297
 - HEADLINE option 297
 - modifying column label for ACROSS variable 311
 - multi-column reports 301–302
 - NOWD option 289–290
 - ordering reports with nonprinting variables 306–307
 - PANELS= option 301–302
 - PRINT procedure and 283, 288, 290
 - producing report breaks 303–306
 - producing summary reports 293–294
 - RBREAK statement 303–306
 - selecting variables for report 291
 - SPLIT= option 294–296
- reports
- See also* customized reports
 - See also* displaying data
 - See also* listings
 - BY statement vs. CLASS statement 324
 - DATA_NULL_ reporting 68
 - detail reports 291–293
 - multi-column 301–302
 - output data sets and 409
 - producing 11–18
 - summary reports 291–294, 303
- RESET=all graphics option 412–413
- restructuring data sets
- with DATA step 494–497
 - with TRANSPOSE procedure 497–500
- RETAIN statement
- assignment statement and 473
 - computing differences between first/last observations 515–516
 - default missing values and 497
 - setting initial values with 121–122
- retained variables 515–517
- return values from observations 204–207
- right joins (SQL) 543–545
- RIGHT option, DEFINE statement (REPORT) 295
- ROUND function 147, 190–191, 201
- rounding numeric values 190–191
- row indices 254
- ROWPCTN keyword 382–383
- ROWPCTSUM keyword 384
- rows
- changing report order 299–300
 - computing percentages 379–380, 384–385
 - crosstab tables and 356–357
 - displaying percentages in 381–382
 - observations and 18, 31, 536
- RTF output destination 402–403
- RTS= option, TABLE statement (TABULATE) 381
- RUN statement
- need for 13
 - SAS processing 24
 - semicolon (;) and 36
- ## S
- _SAME_ keyword 469
- SAMEDAY alignment, INTNX function 156
- SAMPsize= option, SURVEYSELECT procedure 200
- SAS
- getting data into 4
 - inner workings of 22–24
 - overview 3–4
- SAS/ACCESS 88
- SAS Display Manager 9, 406
- SAS Enterprise Guide 9
- SAS Explorer
- conversion process and 98
 - documenting data sets with 80
 - viewing data sets with 60–63
- SAS/GRAPH
- See* graphics
- SAS library 59
- SAS macros 525–527
- SAS names 7–8
- SAS programs

- debugging 68
- enhancing 18–20
- interrupting 134
- producing reports 11–18
- reading raw data 11–18
- sample 4–7
- submitting 14
- writing data lines in 36
- SAS sessions 58
- SAS/STAT 200
- SCAN function 230–232, 306
- scatter plots 425–427
- searching
 - for blanks 225
 - for cases 225–226
 - for character classes 225–226
 - for character values 220–222
 - for characters 220–223
 - for digits 225
 - for punctuation 225
 - for words in strings 223–225
- seed numbers 199
- SEED= option, SURVEYSELECT procedure 200
- SELECT clause (SQL) 537, 539–542
- SELECT statement
 - conditional processing and 108–109
 - FORMAT procedure 84, 488
 - LEAVE statement and 135
 - maintaining formats 477, 479
- semi-colon (;)
 - comment statements and 21
 - DATA step and 36
 - RUN statement and 36
 - SAS programs and 6
- sessions 58
- SET statement 66
 - adding observations to data sets 164–167
 - arrays and 246
 - BY statement and 167–168, 507–508
 - combining detail/summary data 168–169
 - concatenating data sets 546
 - DATA step and 177
 - END= option 445
 - macro variables transferring values 530–531
 - missing character values in arrays 247–248
 - subsetting data sets 163
- single trailing at sign (@) 130, 451–454
- SKIP option, BREAK statement (REPORT) 305
- slash
 - See forward slash
- SMALLEST function 196
- Social Security numbers 180
- SORT procedure
 - changing listing order 270–271
 - DESCENDING option 270–271
 - OUT= option 271
 - PRINT procedure and 299
 - sort flag and 168
 - sorting by multiple variables 272–273
- sorting multiple variables 272–273
- spaces
 - See blanks
- special functions 201–203
- SPEDIS function 234–235, 552
- SPLIT= option, REPORT procedure 294–296
- SPSS 244, 246
- SQL procedure 536–538
 - concatenating data sets 546–549
 - FROM clause 537, 539–542
 - full joins 543–545
 - fuzzy matching 551–552
 - joining tables 539–542
 - left joins 543–545
 - ON clause 543–545
 - ORDER clause 551
 - right joins 543–545
 - SELECT clause 537, 539–542
 - summary functions 549–550
 - UNION operator 546–549
 - WHERE clause 537, 541–542, 552
- SQRT function 127, 197–198
- square brackets [] 245, 480
- ssn11. format 180

- standardizing addresses 236–238
- STAR statement 414
- statements
 - basic rules 6
 - imbedding comments in 20–21
- statistics
 - bar charts representing 420–422
 - computing 14
 - computing row/column percentages 379–380
 - descriptive 328–329, 370–372
 - descriptive statistics functions 194–196
 - displaying with ACROSS variables 311–313
 - grand mean 332
 - in summary reports 293–294
 - naming variables in output data sets 329–330
 - options with MEANS procedure 321
 - outputting summary data sets 330–331
 - outputting with MEANS procedure 328–329
 - RBREAK statement and 303
 - selecting for variables 337–338, 371
 - summary reports and 291
 - t*-tests 407
 - underscore (_) and 329, 337–338
- STD statistic 321
- storing
 - dates 142
 - formats 79
- strings
 - comparing 232–234
 - concatenating 215–217
 - dividing into words 230–232
 - extracting parts of 228–230
 - removing characters from 214–215, 218–220
 - searching for words in 223–225
- STRIP function 217–218
- SUBGROUP= option, VBAR statement (GCHART) 424
- subscripts 245
- subsetting data sets 112, 162–164
- subsetting IF statement 105–107
 - controlling observations with 174
 - LENGTHN function and 213
 - subsetting data sets and 162
- SUBSTR function 228–230
- subtotals
 - adding to listings 274–277
 - producing in reports 303–306
- subtraction in assignment statements 19–20
- SUM function 195–197, 549–550
- sum statement 120–125
 - adding subtotals/totals to listings 274, 276
 - iterative DO loop and 125
- SUM statistic 321, 384, 421
- SUMMARIZE option
 - BREAK statement (REPORT) 305
 - RBREAK statement (REPORT) 303
- summarizing data
 - applying formats to class variables 325–326
 - BY statement with MEANS procedure 323–324, 327
 - CLASS statement with MEANS procedure 324–325, 327
 - creating summary data sets 327–328
 - multiple class variables when 333–337
 - naming variables in output data sets 329–330
 - outputting descriptive statistics 328–329
 - outputting summary data sets 330–333
 - selecting statistics for variables 337–338
 - with MEANS procedure 320–322
- summary data sets
 - creating in DATA step 336–337
 - creating with MEANS procedure 327–328
 - outputting in BY statement 330–331
 - outputting in CLASS statement 331–333
 - selecting statistics for variables 337–338
- SUMMARY procedure
 - creating summary data sets 327–328
 - formats and 463
 - multilabel formats 482

- selecting statistics for variables 337–338
 - summary reports
 - BREAK statement and 303
 - comparing with detail reports 291–293
 - producing 293–294
 - sums, bar charts representing 420–422
 - SUMVAR= option, VBAR statement (GCHART) 421–422
 - SUPPRESS option, BREAK statement (REPORT) 306
 - SURVEYSELECT procedure 200
 - DATA= option 200
 - METHOD= option 200
 - OUT= option 200
 - SAMPSIZE= option 200
 - SEED= option 200
 - swap and drop technique 202, 221
 - SYMBOL statement 412–413
 - connecting points 427–429
 - connecting points with smooth line 429–430
 - INTERPOL= option 427–429
 - JOIN option 428–429
 - LINE= option 429
 - producing scatter plots 425–426
 - VALUE= option 426
 - WIDTH= option 428
 - SYMPUT routine 531
 - &SYSDATE9 macro variable 523
 - &SYSTIME macro variable 523
- T**
- tab character 35
 - table lookup
 - formats and 470–471
 - informats and 470–471
 - INPUTN function 485–490
 - multidimensional arrays for 254–257
 - table of contents 400–401
 - TABLE statement, TABULATE procedure 365
 - asterisk (*) in 368
 - comma in 367
 - concatenation operator and 366
 - descriptive statistics and 370–372
 - missing values and 385
 - MISSTEXT= option 389, 485
 - PRINTMISS option 484–485
 - RTS= option 381
 - tables
 - See also* columns
 - See also* rows
 - See also* TABULATE procedure
 - combining class/analysis variables in 372–373
 - complex 377–378
 - controlling dimensions of 368
 - creating 312
 - crosstab 356–358
 - customizing 374–377
 - data sets and 536
 - joining 539–545
 - merging 539–545
 - observations and 18, 31
 - three-way 358–360
 - two-dimensional 381–382
 - two-way 356–358
 - variables and 18, 31
 - TABLES statement, FREQ procedure 13
 - counting number of visits 509
 - formats and 77
 - MISSING option 351–352
 - MISSPRINT= option 388
 - multiple two-way tables 358
 - NOCOL option 359
 - NOCUM option 345
 - NOPERCENT option 346, 359
 - NOROW option 359
 - producing two-way tables 356–357
 - selecting variables and 345–346
 - tabular reports
 - See* TABULATE procedure
 - TABULATE procedure 364–365
 - See also* TABLE statement, TABULATE procedure
 - ALL keyword 369

- TABULATE procedure (*continued*)
- analysis variables and 372–373, 377–378
 - CLASS statement and 365
 - class variables and 365
 - combining class/analysis variables 372–373
 - complex tables 377–378
 - computing percentages on numeric variables 384–385
 - computing row/column percentages 379–380, 382–383
 - controlling decimal places with 322
 - creating tables 312
 - customizing tables 374–377
 - FORMAT= option 366
 - formats and 463
 - KEYLABEL statement 375–376, 380, 383
 - MISSING option 387–388
 - missing values and 385–389
 - multi-label formats 482–484
 - NOSEPS option 376, 381
 - operators for 366–368
 - percentages in two-dimensional tables 381–382
 - producing descriptive statistics 370–372
- temporary arrays 251–252
- loading initial values into 253
 - table lookups with 254–257
- `_TEMPORARY_` keyword 252, 256
- text wrapping 294–296
- three-way tables 358–360
- TITLE statement 13
- automatic macro variables in 523
 - connecting points and 428
 - displaying data with 268–270
 - quotation marks in 57–58
 - RESET=all graphics option and 412
 - sample SAS program 16
- titles
- adding to listings 268–270
 - font settings in 413
- TODAY function 148–149
- tokens 527–529
- totals
- adding to listings 274–277
 - producing in reports 303–306
- trailing blanks, removing 217–218, 233–234
- transaction files 183–184
- TRANSLATE function 235–237, 256
- TRANSPOSE procedure 497–500
- DROP= option 499
 - PREFIX= option 500
 - RENAME= option 499
- TRANWRD function 235–238
- TRIM function
- NOT functions and 227
 - removing trailing blanks 217–218, 233–234
- truncating numeric values 190–191
- TRUNCOVER option, INFILE statement 143, 443
- TTEST procedure 407–408
- t*-tests 407–408
- t*-values 407–409
- two-digit years 145
- two-dimensional tables 381–382
- two-way tables 356–358
- TYPE= option, VBAR statement (GCHART) 421–422
- `_TYPE_` variable 332–337
- ## U
- UL option, RBREAK statement (REPORT) 303
- underscore (`_`)
- as wildcard 114
 - conversion process and 92, 96
 - naming conventions and 7
 - statistics and 329, 337–338
- UNION ALL CORRESPONDING operator 546
- UNION ALL operator 546
- UNION CORRESPONDING operator 546
- UNION operator (SQL) 546–549
- UNIVARIATE procedure 403–405
- UPCASE function 214, 235

- UPCASE option, INVALUE statement (FORMAT) 466, 469
- UPDATE statement 183–184
- uppercase 214, 235
- user-defined formats 79–82, 380, 464
- user-defined informats 464–467
- V**
- VALUE= option, SYMBOL statement (GPLOT) 426
- VALUE statement, FORMAT procedure 74, 78, 482–485
 - MULTILABEL option 482–485
- VAR statement 14
 - changing listing appearance with 263–265
 - descriptive statistics and 370
 - double dash in 149
 - ID statement and 75
 - MEANS procedure and 322, 328–330
 - TABLE statement (TABULATE) and 365
- VAR statistic 321
- variable lists 149, 455–456
- variable names
 - array references and 245
 - defined 13
 - in INPUT statement 31
 - informats and 43–44
- variables
 - See also* character variables
 - See also* class variables
 - See also* macro variables
 - See also* numeric variables
 - adding labels to 71–73
 - adding to bar charts 423–425
 - analysis variables 292–295, 312, 372–373, 377–378
 - applying ORDER usage to 300–301
 - changing order in COLUMN statement 297–298
 - computing frequencies of 342–344
 - computing with REPORT procedure 307–308
 - continuous 416–420
 - controlling decimal places 322
 - controlling listing appearance 263–265
 - creating 249–250, 463–464
 - defining usage for 296
 - _FREQ_ 331, 337
 - grouping 296–297
 - in ID statements 75
 - listing formats 80
 - listing labels 80
 - missing values in TABULATE procedure 385–389
 - naming conventions 7
 - naming in output data sets 329–330
 - nonprinting 306–307
 - recoding with formats 462–463
 - retained 515–517
 - selecting for FREQ procedure 345–346
 - selecting for reports 291
 - selecting statistics for 337–338, 371
 - setting initial values for 121–122
 - sorting by multiple 272–273
 - sum statement and 123
 - swap and drop technique 202
 - table columns and 18, 31, 536
 - _TYPE_ 332–337
 - types of 8
 - VAR statement and 149
 - WHERE statement and 162
- VARNUM option, CONTENTS procedure 58, 149
- VAXIS option, PLOT statement (GPLOT) 426
- VBAR statement, GCHART procedure 414
 - DISCRETE option 419–420
 - GROUP= option 423
 - MAXIS= option 421
 - MIDPOINTS= option 417–418
 - SUBGROUP= option 424
 - SUMVAR= option 421–422
 - TYPE= option 421–422
- VBAR3D statement 414
- VERIFY function 227–228
- VIEWTABLE Window 64–65

virtual data sets 474
 visits, counting number of 509–512
 VSIZE=4 option, GOPTIONS statement 414

W

\$w. informat 40
 w.d informat 40
 WEEKDAY function 149, 419
 WHEN statement 108–109
 WHERE clause (SQL) 537
 fuzzy matching 552
 joining tables 541–542
 WHERE= data set option 499
 WHERE statement
 controlling observation appearance in
 listings 266–267
 subsetting data sets 112, 162
 TYPE variable in 336
 useful operators 113–114
 WIDTH= option, SYMBOL statement 428
 wildcards
 asterisk (*) as 338, 446, 538
 colon (:) as 202, 337
 for WHERE statement operators 114
 question mark as 446
 Williams, Christianna 536
 words
 dividing strings into 230–232
 searching for in strings 223–225
 substituting 235–238
 wrapping lines of text 294–296

X

XLS engine 95–96
 X-Y plots 425–427

Y

YEAR function 149
 YEARCUTOFF system option 145, 150
 years
 computing between dates 146–147
 creating dates from 150–151
 date interval functions 152–157

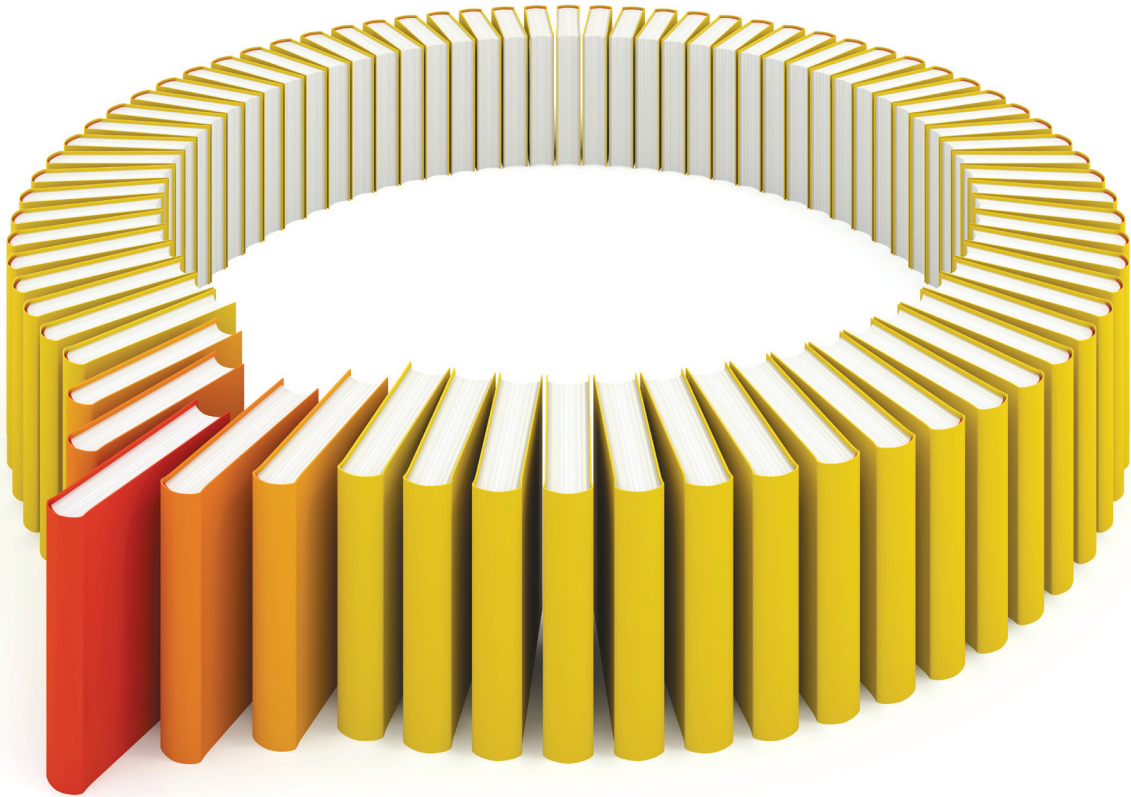
 extracting from dates 149–150
 four-digit 145
 two-digit 145
 YRDIF function 146–147

Z

Zdeb, Mike 180

Symbols

& (ampersand) 46
 * (asterisk)
 See asterisk (*)
 @ (at sign)
 See at sign (@)
 @@ (double trailing @ sign) 197, 454–455
 : (colon)
 See colon (:)
 , (comma)
 See comma (,)
 { } (curly brackets) 245, 254
 \$ (dollar sign)
 See dollar sign (\$)
 = (equal sign)
 See equal sign (=)
 / (forward slash)
 See forward slash (/)
 - (hyphen) 180
 < (less than sign) 78
 () (parentheses)
 See parentheses ()
 % (percent sign) 114, 522
 . (period)
 See period (.)
 + (plus sign) 123, 456–457
 ? (question mark) 247–248, 446
 " (quotation marks)
 See quotation marks (")
 ; (semicolon)
 See semicolon (;)
 [] (square brackets) 245, 480



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

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

 support.sas.com/bookstore
for additional books and resources.


THE POWER TO KNOW[®]