THE POWER TO KNOW.

SAS® Certification Prep Guide: Base Programming for SAS®9, Fourth Edition

SAS® Documentation
December 6, 2017

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS® Certification Prep Guide: Base Programming for SAS®9, Fourth Edition*. Cary, NC: SAS Institute Inc.

SAS® Certification Prep Guide: Base Programming for SAS®9, Fourth Edition

Copyright © 2017, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-63526-373-2 (Hard copy)
ISBN 978-1-63526-468-5 (Epub)
ISBN 978-1-63526-469-2 (Mobi)
ISBN 978-1-63526-470-8 (PDF)

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

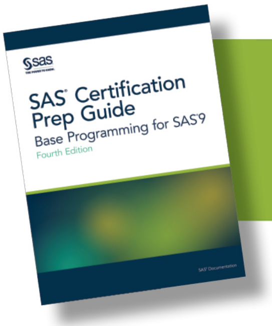
SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

January 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

P2:certpgbp



An Introduction to SAS® Certification Guide: Base Programming for SAS®9, Fourth Edition. Full book available for purchase here.

Contents

How to Prepare for the SAS Base Programming for SAS®9 Exam . . . . *ix*
Accessibility Features of the SAS Certification Prep Guide . . . . *xiii*

Chapter 1 • Setting Up Practice Data **1**
Determine What SAS Solution or Environment You Are Using 1
Determine Whether You Have Write Access 1
SAS Windowing Environment 2
SAS Studio and SAS University Edition 2
SAS Enterprise Guide 3

Chapter 2 • Basic Concepts **5**
Getting Started 6
The Basics of the SAS Language 6
SAS Libraries 11
Referencing SAS Files 13
SAS Data Sets 15
Chapter Quiz 22

Chapter 3 • Setting Up Your SAS Session **27**
SAS Libraries 28
Viewing SAS Libraries 31
Specifying Results Formats 36
Setting System Options 41
Chapter Quiz 48

Chapter 4 • Identifying and Correcting Errors **51**
Error Messages 51
Correcting Common Errors 53
Chapter Quiz 62

Chapter 5 • Creating List Reports **67**
Creating a Basic Report 68
Selecting Variables 69
Identifying Observations 70
Sorting Data 74
Generating Column Totals 76
Double Spacing LISTING Output 82
Specifying Titles and Footnotes in Procedure Output 82
Assigning Descriptive Labels 88
Formatting Data Values 90
Using Permanently Assigned Labels and Formats 93
Chapter Quiz 94

Chapter 6 • Creating SAS Data Sets from External Files **99**
Raw Data Files 100
Creating a SAS Data Set from a Raw Data File 101
Referencing a Raw Data File 102
DATA Step Statements for Reading Data 104
Reading and Verifying the Data 108

Creating and Modifying Variables 111
Subsetting Data 115
Reading Instream Data 116
Creating a Raw Data File 118
Reading Microsoft Excel Data 121
LIBNAME Statement Options 129
Creating Excel Worksheets 130
The IMPORT Procedure 130
Chapter Quiz 134

***Chapter 7 • Understanding DATA Step Processing* 139**
How SAS Processes Programs 139
Compilation Phase 141
Execution Phase 144
Debugging a DATA Step 152
Testing Your Programs 158
Chapter Quiz 161

***Chapter 8 • Creating and Applying User-Defined Formats* 165**
The FORMAT Procedure 165
Defining a Unique Format 167
Associating User-Defined Formats with Variables 169
Chapter Quiz 172

***Chapter 9 • Producing Descriptive Statistics* 175**
Computing Statistics Using PROC MEANS 176
Selecting Statistics 177
Limiting Decimal Places 180
Specifying Variables in PROC MEANS 181
Group Processing Using the CLASS Statement 182
Group Processing Using the BY Statement 183
Creating a Summarized Data Set Using PROC MEANS 184
Creating a Summarized Data Set Using PROC SUMMARY 186
Producing Frequency Tables Using PROC FREQ 187
Specifying Variables in PROC FREQ 189
Creating Two-Way Tables 191
Creating N-Way Tables 192
Creating Tables in List Format 194
Chapter Quiz 198

***Chapter 10 • Creating Output with ODS* 203**
The Output Delivery System (ODS) 204
Creating HTML Output with ODS 206
Creating PDF Output with ODS 215
Creating RTF Output with ODS 220
Creating EXCEL Output with ODS 222
Chapter Quiz 228

***Chapter 11 • Creating and Managing Variables* 231**
Creating and Modifying Variables 232
Assigning Values Conditionally 234
Specifying Lengths for Variables 238
Subsetting Data 240
Assigning Permanent Labels and Formats 243
Assigning Values Conditionally Using SELECT Groups 245
Grouping Statements Using DO Groups 248

Chapter Quiz 250

Chapter 12 • Reading SAS Data Sets 255

Reading a Single Data Set to Create Another 255
Manipulating Data 257
Using BY-Group Processing 259
Writing Observations Explicitly 263
Detecting the End of a Data Set 264
Chapter Quiz 273

Chapter 13 • Combining SAS Data Sets 277

How to Prepare Your Data Sets 278
Methods of Combining SAS Data Sets: The Basics 278
One-to-One Reading: Details 280
Concatenating: Details 284
Appending: Details 286
Interleaving: Details 289
Match-Merging: Details 291
Match-Merge Processing 295
Renaming Variables 300
Excluding Unmatched Observations 302
Selecting Variables 304
Chapter Quiz 305

Chapter 14 • Using Functions to Manipulate Data 313

The Basics of SAS Functions 314
SAS Functions Syntax 315
Converting Data with Functions 316
Manipulating SAS Date Values with Functions 323
Modifying Character Values with Functions 340
Modifying Numeric Values with Functions 360
Nesting SAS Functions 362
Chapter Quiz 362

Chapter 15 • Processing Data with DO Loops 367

The Basics of DO Loops 367
Constructing DO Loops 370
Nesting DO Loops 373
Iteratively Processing Observations from a Data Set 375
Conditionally Executing DO Loops 376
Using Conditional Clauses with the Iterative DO Statement 377
Chapter Quiz 379

Chapter 16 • Processing Data with Arrays 383

The Basics of SAS Arrays 383
Creating One-Dimensional Arrays 384
Expanding Your Use of Arrays 391
Chapter Quiz 397

Chapter 17 • Reading Raw Data in Fixed Fields 401

Identifying the Type of Numeric Data 401
Using Formatted Input 403
Using Informats 406
Using Record Formats 410
Chapter Quiz 413

Chapter 18 • Reading and Creating Raw Data in Free-Format **417**
Free-Format Data 417
Reading Standard Data with List Input 418
Specifying the Length of Character Variables 421
Reading Free-Format Data with Non-Blank Delimiters and Missing Values 424
Reading Nonstandard Data with List Input 431
Creating Free-Format Data 436
Write a Comma-Delimited File Using Formats 439
Chapter Quiz 441

Chapter 19 • SAS Date and Time Values **447**
Reading Dates and Times with Informats 447
Using Dates and Times in Calculations 451
Displaying Date and Time Values with Formats 456
Chapter Quiz 458

Chapter 20 • Creating a Single Observation from Multiple Raw Data Records **461**
A Word About Creating a Single Observation 461
Using Line Pointer Controls 462
Reading Multiple Records Sequentially 462
Reading Multiple Records Non-Sequentially 469
Combining Line Pointer Controls 473
Chapter Quiz 474

Chapter 21 • Creating Multiple Observations from a Single Raw Data Record **479**
A Word About Creating Multiple Observations 479
Reading Repeating Blocks of Data 480
Reading the Same Number of Repeating Fields 487
Reading a Varying Number of Repeating Fields 495
Chapter Quiz 501

Chapter 22 • Conditioning Raw Data Files **507**
A Word About Conditioning Raw Data Files 507
Creating One Observation per Detail Record 508
Creating One Observation per Header Record 515
Processing a DATA Step That Creates One Observation per Header Record 520
Chapter Quiz 522

Appendix 1 • Quiz Answer Keys **527**
Chapter 2: Basic Concepts 527
Chapter 3: Setting Up Your SAS Session 528
Chapter 4: Identifying and Correcting Errors 529
Chapter 5: Creating List Reports 531
Chapter 6: Creating SAS Data Sets from External Files 532
Chapter 7: Understanding DATA Step Processing 533
Chapter 8: Creating and Applying User-Defined Formats 534
Chapter 9: Producing Descriptive Statistics 535
Chapter 10: Creating Output with ODS 536
Chapter 11: Creating and Managing Variables 537
Chapter 12: Reading SAS Data Sets 538
Chapter 13: Combining SAS Data Sets 539
Chapter 14: Using Functions to Manipulate Data 541
Chapter 15: Processing Data with DO Loops 542
Chapter 16: Processing Data with Arrays 542
Chapter 17: Reading Raw Data in Fixed Fields 543
Chapter 18: Reading and Creating Raw Data in Free-Format 544

Chapter 19: SAS Date and Time Values 545
Chapter 20: Creating a Single Observation from Multiple Raw Data Records 546
Chapter 21: Creating Multiple Observations from a Single Raw Data Record 547
Chapter 22: Conditioning Raw Data Files 548

Index **. 551**

An Introduction to SAS® Certification Guide: Base Programming for SAS®9, Fourth Edition. Copyright © 2017, SAS Institute Inc., Cary, North Carolina, USA. ALL RIGHTS RESERVED.

How to Prepare for the SAS Base Programming for SAS®9 Exam

Requirements and Details

Requirements

To complete examples in this book, you must have access to Base SAS, SAS Enterprise Guide, or SAS Studio. See Chapter 1, “Setting Up Practice Data,” to ensure you have proper access.

Exam Objectives and Updates to This Book

The current exam objectives and a list of any updates to this book are available at www.sas.com/certify. Exam objectives are subject to change.

Take a Practice Exam

Practice exams are available for purchase through SAS and Pearson VUE. For more information about practice exams, see www.sas.com/base_programmer_cert.

Registering for the Exam

To register for the official SAS Base Programming for SAS®9 exam, see the SAS Global Certification website at www.sas.com/certify.

Additional Resources for Learning SAS Programming

From SAS Software	
Help	• SAS®9: Select **Help** ⇨ **SAS Help and Documentation**. • SAS Enterprise Guide: Select **Help** ⇨ **SAS Enterprise Guide Help**. • SAS Studio: Select the Help icon ? .
Documentation	• SAS®9: Select **Help** ⇨ **SAS Help and Documentation**. • SAS Enterprise Guide: Access online documentation on the web. • SAS Studio: Select the Help icon ? and then click **Help**.

On the Web	
Base SAS Glossary	support.sas.com/baseglossary
Bookstore	www.sas.com/books
Certification	www.sas.com/certify
Community	communities.sas.com
Knowledge Base	support.sas.com/notes
Learning Center	www.sas.com and click **Learn**. Then select **Get Started Learning SAS**.
SAS Documentation	support.sas.com/documentation documentation.sas.com
SAS Global Academic Program	www.sas.com and click **Learn**. Then select **For Students and Educators**.
SAS OnDemand	support.sas.com/ondemand/
Training	www.sas.com/training
Technical Support	support.sas.com. Then select **Technical Support**.

Syntax Conventions

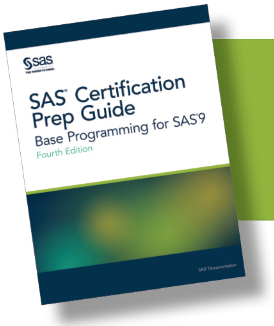
In this book, SAS syntax looks like this example:

DATA *output-SAS-data-set*
        (**DROP**=*variables(s)* | **KEEP**=*variables(s)*);
    SET *SAS-data-set* <*options*>;
    BY *variable(s)*
RUN;

Here are the conventions that are used in the example:

- DATA, DROP=, KEEP=, SET, BY, and RUN are in uppercase bold because they must be spelled as shown.
- *output-SAS-data-set*, *variable(s)*, *SAS-data-set*, and *options* are in italics because each represents a value that you supply.
- <*options*> is enclosed in angle brackets because it is optional syntax.
- DROP= and KEEP= are separated by a vertical bar (|) to indicate that they are mutually exclusive.

The example syntax that is shown in this book includes only what you need to know in order to prepare for the certification exam. For complete syntax, see the appropriate SAS reference guide.



SAS Certification Prep Guide
Base Programming for SAS9
Fourth Edition

An Introduction to SAS® Certification Guide: Base Programming for SAS®9, Fourth Edition. Full book available for purchase here.

Chapter 2

Basic Concepts

Getting Started 6

The Basics of the SAS Language 6
- SAS Statements 6
- Global Statements 6
- DATA Step 6
- PROC Step 7
- A Simple SAS Program 7
- Processing SAS Programs 8
- Log Messages 8
- Results of Processing 9

SAS Libraries 11
- Definition 11
- Predefined SAS Libraries 11
- Defining Libraries 11
- How SAS Files Are Stored 12
- Storing Files Temporarily or Permanently 12

Referencing SAS Files 13
- Referencing Permanent SAS Data Sets 13
- Referencing Temporary SAS Files 14
- Rules for SAS Names 14

SAS Data Sets 15
- Overview of Data Sets 15
- Descriptor Portion 15
- Data Portion 16
- Variable Attributes 17
- Variable Names 18
- Type 19
- Length 20
- Format 20
- Informat 21
- Label 21
- SAS Indexes 22
- Extended Attributes 22

Chapter Quiz 22

Getting Started

In the SAS Base Programming for SAS®9 exam, you are not tested on the details of running SAS software in the various environments. However, you might find such information useful when working with the practice data.

You can access a brief overview on the windows and menus of the SAS windowing environment, Enterprise Guide, and SAS Studio at http://video.sas.com/. From **Categories** select **How To Tutorials** ⇨ **Programming**. Select the video for your SAS environment. Other tutorials are available from the SAS website.

The Basics of the SAS Language

SAS Statements

A SAS statement is a type of SAS language element that is used to perform a particular operation in a SAS program or to provide information to a SAS program. SAS statements are free-format. This means that they can begin and end anywhere on a line, that one statement can continue over several lines, and that several statements can be on the same line. Blank or special characters separate words in a SAS statement.

TIP You can specify SAS statements in uppercase or lowercase. In most situations, text that is enclosed in quotation marks is case sensitive.

Here are two important rules for writing SAS programs:

- A SAS statement ends with a semicolon.
- A statement usually begins with a SAS keyword.

There are two types of SAS statements:

- statements that are used in DATA and PROC steps
- statements that are global in scope and can be used anywhere in a SAS program

Global Statements

Global statements are used anywhere in a SAS program and stay in effect until changed or canceled, or until the SAS session ends. Here are some common global statements: TITLE, LIBNAME, OPTIONS, and FOOTNOTE.

DATA Step

The DATA step creates or modifies data. The input for a DATA step can be of several types, such as raw data or a SAS data set. The output from a DATA step can be of several types, such as a SAS data set or a report. A SAS data set is a data file that is formatted in a way that SAS can understand.

For example, you can use DATA steps to do the following:

- put your data into a SAS data set

- compute values
- check for and correct errors in your data
- produce new SAS data sets by subsetting, supersetting, merging, and updating existing data sets

PROC Step

The PROC (procedure) step analyzes data, produces output, or manages SAS files. The input for a PROC step is usually a SAS data set. The output from a PROC step can be of several types, such as a report or an updated SAS data set.

For example, you can use PROC steps to do the following:

- create a report that lists the data
- produce descriptive statistics
- create a summary report
- produce plots and charts

A Simple SAS Program

This program uses an existing SAS data set to create a new SAS data set containing a subset of the original data set. It then prints a listing of the new data set using PROC PRINT.

```
data sasuser.admit2;
   set sasuser.admit;
   where age>39;
run;
proc print data=sasuser.admit2;
run;
```

The sample SAS program contains a DATA step and a PROC step. The DATA step produced a new SAS data set. Only those observations with an age value greater than 39 are written to the new SAS data set.

A DATA step begins with a DATA statement, which begins with the keyword DATA. A PROC step begins with a PROC statement, which begins with the keyword PROC. The sample program contains the following statements:

Table 2.1 *SAS Program Statements*

Statements	Sample Program Code
DATA statement	`data sasuser.admit2;`
SET statement	`set sasuser.admit;`
Additional programming statements	`where age>39;`
RUN statement	`run;`
PROC PRINT statement	`proc print data=sasuser.admit2;`

Statements	Sample Program Code
RUN statement	`run;`

Processing SAS Programs

When you submit a SAS program, SAS begins reading the statements and checking them for errors.

DATA and PROC statements signal the beginning of a new step. The RUN statement (for most procedures and the DATA step) and the QUIT statement (for some procedures) mark step boundaries. The beginning of a new step (DATA or PROC) also implies the end of the previous step. At a step boundary, SAS executes any statements that have not previously executed and ends the step. In the sample program, each step ends with a RUN statement.

```
data sasuser.admit2;
   set sasuser.admit;
   where age>39;
run;
proc print data=sasuser.admit2;
run;
```

TIP The RUN statement is not required between steps in a SAS program. It is a best practice to use a RUN statement because it can make the SAS program easier to read and the SAS log easier to understand when debugging.

Log Messages

The SAS log collects messages about the processing of SAS programs and about any errors that occur. Each time a step is executed, SAS generates a log of the processing activities and the results of the processing.

When SAS processes the sample program, it produces the log messages shown below. Notice that you get separate sets of messages for each step in the program.

Log 2.1 *SAS Log Messages for Each Program Step*

```
5    data sasuser.admit2;
6        set sasuser.admit;
7        where age>39;
8    run;

NOTE: There were 10 observations read from the data set SASUSER.ADMIT.
      WHERE age>39;
NOTE: The data set SASUSER.ADMIT2 has 10 observations and 9 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

9    proc print data=sasuser.admit2;
NOTE: Writing HTML Body file: sashtml.htm
10   run;

NOTE: There were 10 observations read from the data set SASUSER.ADMIT2.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.35 seconds
      cpu time            0.24 seconds
```

Results of Processing

The DATA Step

Suppose you submit the sample program below:

```
data sasuser.admit2;
    set sasuser.admit;
    where age>39;
run;
```

When the program is processed, it creates a new SAS data set (sasuser.admit2) containing only those observations with age values greater than 39. The DATA step creates a new data set and produces messages in the SAS log, but it does not create a report or other output.

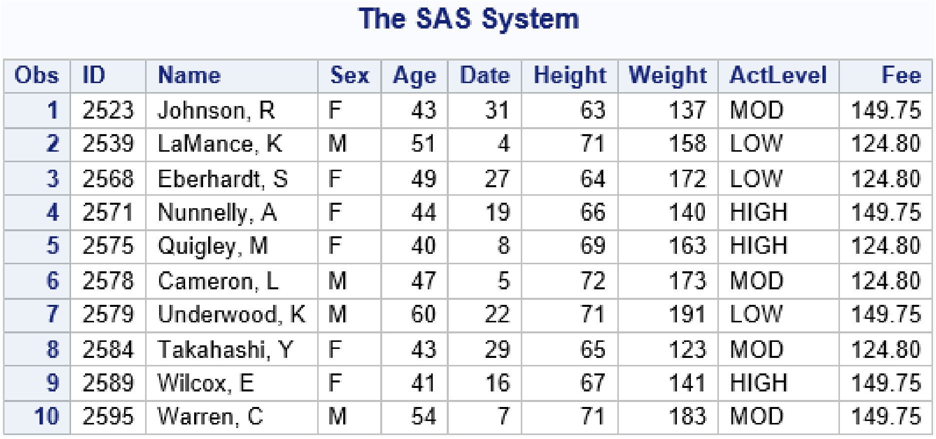
The PROC Step

If you add a PROC PRINT step to this same example, the program produces the same new data set as before, but it also creates the following report, which is displayed in HTML:

```
data sasuser.admit2;
    set sasuser.admit;
    where age>39;
run;
proc print data=sasuser.admit2;
run;
```

Note: The default output in SAS Enterprise Guide is SAS Report. To change the default output in SAS Enterprise Guide to HTML, click **Tools** and select **Options** ⇨ **Results** ⇨ **Results General**. Then select HTML. Ensure that you have cleared SAS Report.

Figure 2.1 PRINT Procedure Output



The SAS System

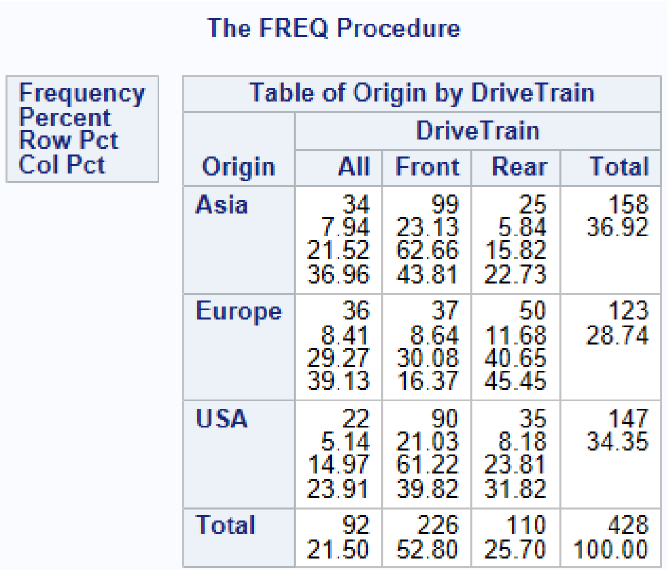
Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2523	Johnson, R	F	43	31	63	137	MOD	149.75
2	2539	LaMance, K	M	51	4	71	158	LOW	124.80
3	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80
4	2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75
5	2575	Quigley, M	F	40	8	69	163	HIGH	124.80
6	2578	Cameron, L	M	47	5	72	173	MOD	124.80
7	2579	Underwood, K	M	60	22	71	191	LOW	149.75
8	2584	Takahashi, Y	F	43	29	65	123	MOD	124.80
9	2589	Wilcox, E	F	41	16	67	141	HIGH	149.75
10	2595	Warren, C	M	54	7	71	183	MOD	149.75

Other Procedures

SAS programs often invoke procedures that create output in the form of a report, as is the case with the FREQ procedure:

```
proc freq data=sashelp.cars;
  table origin*DriveTrain;
run;
```

Figure 2.2 FREQ Procedure Output



The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of Origin by DriveTrain				
	DriveTrain			
Origin	All	Front	Rear	Total
Asia	34 7.94 21.52 36.96	99 23.13 62.66 43.81	25 5.84 15.82 22.73	158 36.92
Europe	36 8.41 29.27 39.13	37 8.64 30.08 16.37	50 11.68 40.65 45.45	123 28.74
USA	22 5.14 14.97 23.91	90 21.03 61.22 39.82	35 8.18 23.81 31.82	147 34.35
Total	92 21.50	226 52.80	110 25.70	428 100.00

Other SAS programs perform tasks such as sorting and managing data, which have no visible results except for messages in the log. (All SAS programs produce log messages, but some SAS programs produce only log messages.)

```
proc copy in=sasuser out=work;
   select admit;
run;
```

Log 2.2 *SAS Log: COPY Procedure Output*

```
11   proc copy in=sasuser out=work;
12       select admit;
13   run;

NOTE: Copying SASUSER.ADMIT to WORK.ADMIT (memtype=DATA).
NOTE: There were 21 observations read from the data set SASUSER.ADMIT.
NOTE: The data set WORK.ADMIT has 21 observations and 9 variables.
NOTE: PROCEDURE COPY used (Total process time):
      real time           0.02 seconds
      cpu time            0.01 seconds
```

SAS Libraries

Definition

A SAS library contains one or more files that are defined, recognized, and accessible by SAS, and that are referenced and stored as a unit. One special type of file is called a *catalog*. In SAS libraries, catalogs function much like subfolders for grouping other members.

Predefined SAS Libraries

By default, SAS defines several libraries for you:

Sashelp
: a permanent library that contains sample data and other files that control how SAS works at your site. This is a Read-Only library.

Sasuser
: a permanent library that contains SAS files in the Profile catalog and that stores your personal settings. This is also a convenient place to store your own files.

  Note: If you are using SAS Studio or SAS University Edition, you might not have Write access to the Sasuser directory. To verify whether you have Write access, see “Determine Whether You Have Write Access” on page 1.

Work
: a temporary library for files that do not need to be saved from session to session.

You can also define additional libraries. When you define a library, you indicate the location of your SAS files to SAS. After you define a library, you can manage SAS files within it.

Note: If you are using SAS Studio, you might encounter Webwork library. Webwork is the default output library in interactive mode. For more information about the Webwork library, see *SAS Studio: User's Guide*.

Defining Libraries

To define a library, you assign a library name to it and specify the location of the files, such as a directory path.

You can also specify an engine, which is a set of internal instructions that SAS uses for writing to and reading from files in a library.

You can define SAS libraries using programming statements. For information about how to write LIBNAME statements to define SAS libraries, see Assigning Librefs on page 28.

TIP Depending on your operating environment and the SAS/ACCESS products that you license, you can create libraries with various engines. Each engine enables you to read a different file format, including file formats from other software vendors.

When you delete a SAS library, the pointer to the library is deleted, and SAS no longer has access to the library. However, the contents of the library still exist in your operating environment.

How SAS Files Are Stored

A SAS library is the highest level of organization for information within SAS.

For example, in the Windows and UNIX environments, a library is typically a group of SAS files in the same folder or directory.

The table below summarizes the implementation of SAS libraries in various operating environments.

Table 2.2 *Environments and SAS Libraries*

Environment	Library Definition
Windows, UNIX	a group of SAS files that are stored in the same directory. Other files can be stored in the directory, but only the files that have SAS file extensions are recognized as part of the SAS library.
z/OS	a specially formatted host data set in which only SAS files are stored.

Storing Files Temporarily or Permanently

Depending on the library name that you use when you create a file, you can store SAS files temporarily or permanently.

Table 2.3 *Temporary and Permanent SAS Libraries*

Temporary SAS libraries last only for the current SAS session.	If you do not specify a library name when you create a file (or if you specify the library name Work), the file is stored in the temporary SAS library. When you end the session, the temporary library and all of its files are deleted.

Permanent SAS libraries are available to you during subsequent SAS sessions.	To store files permanently in a SAS library, specify a library name other than the default library name Work. For example, by specifying the library name **`sasuser`** when you create a file, you specify that the file is to be stored in a permanent SAS library until you delete it.

Referencing SAS Files

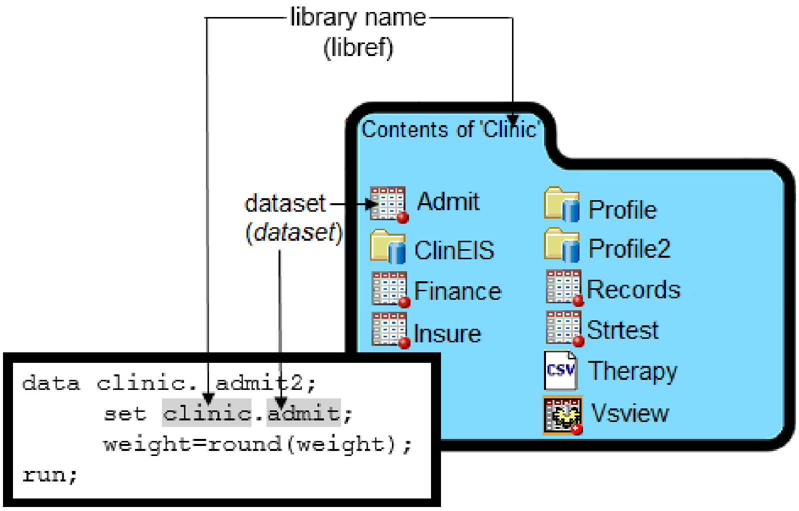
Referencing Permanent SAS Data Sets

To reference a permanent SAS data set in your SAS programs, use a two-level name consisting of the library name and the data set name:

libref.dataset

In the two-level name, *libref* is the name of the SAS library that contains the data set, and *dataset* is the name of the SAS data set. A period separates the libref and data set name.

Figure 2.3 *Two-Level Permanent SAS Name*



```
data clinic. admit2;
      set clinic.admit;
      weight=round(weight);
run;
```

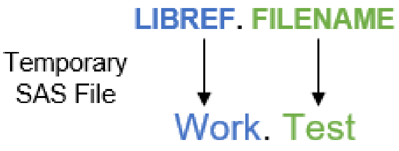
For example, suppose you want to create a new permanent SAS library named Clinic. In the sample program, Clinic.Admit is the two-level name for the SAS data set Admit, which is stored in the library named Clinic.

```
data clinic.admit2;
   set clinic.admit;
   weight =round(weight);
run;
```

Referencing Temporary SAS Files

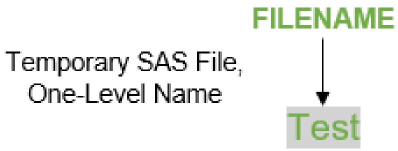
To reference temporary SAS files, you can specify the default libref Work, a period, and the data set name. For example, the two-level name, Work.Test, references the SAS data set named Test that is stored in the temporary SAS library Work.

Figure 2.4 Two-Level Temporary SAS Library Name



Alternatively, you can use a one-level name (the data set name only) to reference a file in a temporary SAS library. When you specify a one-level name, the default libref Work is assumed. For example, the one-level name Test also references the SAS data set named Test that is stored in the temporary SAS library Work.

Figure 2.5 One-Level Temporary SAS Library Name



Rules for SAS Names

The following rules apply to the names of SAS data sets, variables, and libraries:

- They must begin with a letter (A-Z, either uppercase or lowercase) or an underscore (_).
- They can continue with any combination of numbers, letters, or underscores.
- They can be 1 to 32 characters long.
- SAS library names (librefs) can be 1 to 8 characters long.

These are examples of valid data set names and variable names:

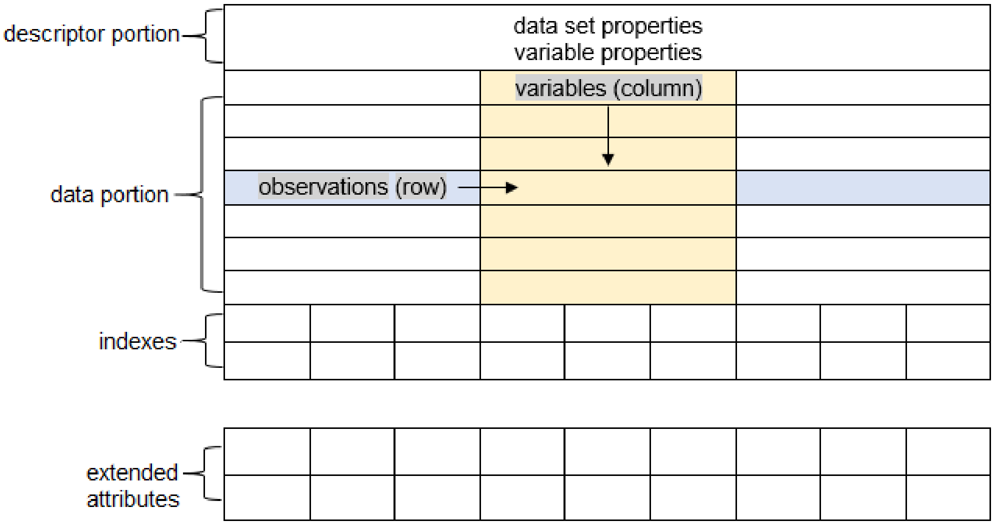
- Payroll
- LABDATA1995_1997
- _EstimatedTaxPayments3

SAS Data Sets

Overview of Data Sets

A SAS data set is a file that consists of two parts: a descriptor portion and a data portion. Sometimes a SAS data set also points to one or more indexes, which enable SAS to locate rows in the data set more efficiently. (The data sets that are shown in this chapter do not contain indexes.) Extended attributes are user-defined attributes that further define a SAS data set.

Figure 2.6 *Parts of a SAS Data Set*



Descriptor Portion

The descriptor portion of a SAS data set contains information about the data set, including the following:

- the name of the data set
- the date and time that the data set was created
- the number of observations
- the number of variables

The table below lists part of the descriptor portion of the data set sasuser.insure, which contains insurance information for patients who are admitted to a wellness clinic.

Table 2.4 *Descriptor Portion of Attributes in a SAS Data Set*

Data Set Name:	SASUSER.INSURE
Member Type:	DATA

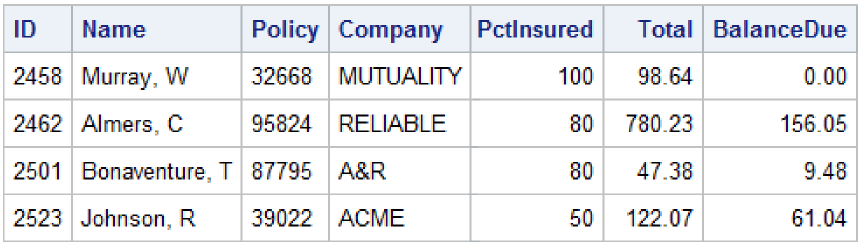
Engine:	V9
Created:	10:05 Thursday, February 16, 2017
Observations:	21
Variables:	7
Indexes:	0
Observation Length:	64

Data Portion

Data Portion Overview

The data portion of a SAS data set is a collection of data values that are arranged in a rectangular table. In the example below, the company **MUTUALITY** is a data value, Policy **32668** is a data value, and so on.

Figure 2.7 Parts of a SAS Data Set: Data Portion

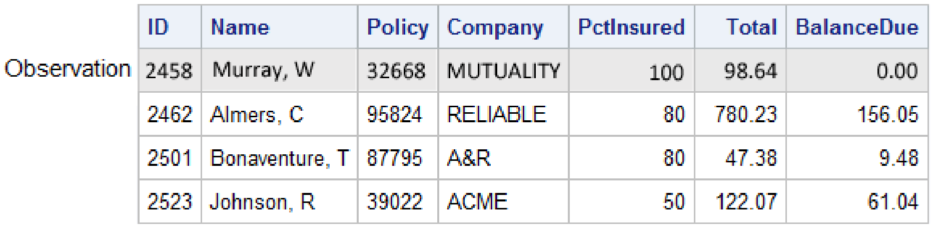


ID	Name	Policy	Company	PctInsured	Total	BalanceDue
2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2501	Bonaventure, T	87795	A&R	80	47.38	9.48
2523	Johnson, R	39022	ACME	50	122.07	61.04

Observations (Rows)

Observations (also called rows) in the data set are collections of data values that usually relate to a single object. The values **2458**, **Murray W**, **32668**, **MUTUALITY**, **100**, **98.64**, and **0.00** comprise a single observation in the data set shown below.

Figure 2.8 Parts of a SAS Data Set: Observations



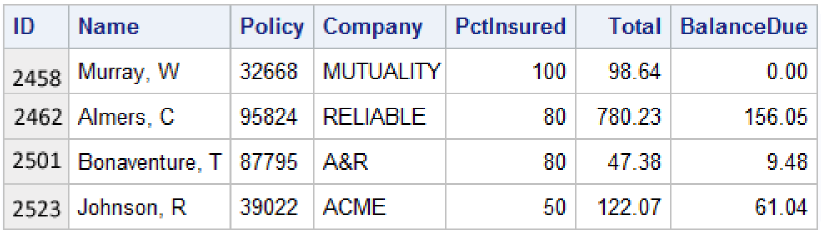
	ID	Name	Policy	Company	PctInsured	Total	BalanceDue
Observation	2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
	2462	Almers, C	95824	RELIABLE	80	780.23	156.05
	2501	Bonaventure, T	87795	A&R	80	47.38	9.48
	2523	Johnson, R	39022	ACME	50	122.07	61.04

This data set has 21 observations, each containing information about an individual. To view the full descriptor portion of this data set, see Table 2.4 on page 15. A SAS data set can store any number of observations.

Variables (Columns)

Variables (also called columns) in the data set are collections of values that describe a particular characteristic. The values `2458`, `2462`, `2501`, and `2523` comprise the variable ID in the data set shown below.

Figure 2.9 Parts of a SAS Data Set: Variables



Variables

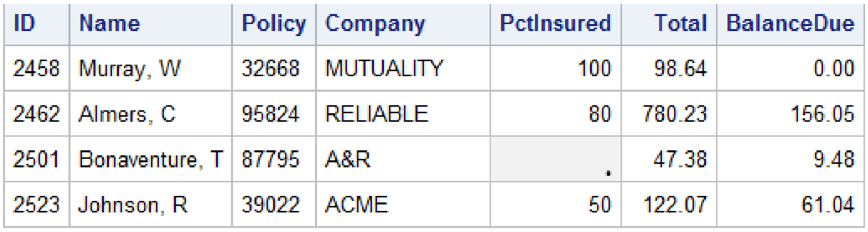
ID	Name	Policy	Company	PctInsured	Total	BalanceDue
2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2501	Bonaventure, T	87795	A&R	80	47.38	9.48
2523	Johnson, R	39022	ACME	50	122.07	61.04

This data set contains seven variables: ID, Name, Policy, Company, PctInsured, Total, and BalanceDue. A SAS data set can store thousands of variables.

Missing Values

Every variable and observation in a SAS data set must have a value. If a data value is unknown for a particular observation, a missing value is recorded in the SAS data set. A period (.) is the default value for a missing numeric value, and a blank space is the default value for a missing character value.

Figure 2.10 Parts of a SAS Data Set: Missing Data Values



ID	Name	Policy	Company	PctInsured	Total	BalanceDue
2458	Murray, W	32668	MUTUALITY	100	98.64	0.00
2462	Almers, C	95824	RELIABLE	80	780.23	156.05
2501	Bonaventure, T	87795	A&R	.	47.38	9.48
2523	Johnson, R	39022	ACME	50	122.07	61.04

Missing Value

Variable Attributes

In addition to general information about the data set, the descriptor portion contains information about the properties of each variable in the data set. The properties information includes the variable's name, type, length, format, informat, and label.

When you write SAS programs, it is important to understand the attributes of the variables that you use. For example, you might need to combine SAS data sets that contain same-named variables. In this case, the variables must be the same type (character or numeric). If the same-named variables are both character variables, you still need to check that the variable lengths are the same. Otherwise, some values might be truncated.

Here is a partial listing of the attribute information in the descriptor portion of the SAS data set sasuser.insure.

Table 2.5 *Variable Attributes in the Descriptor Portion of a SAS Data Set sasuser.insure*

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Variable Names

Rules for Variable Names

Each variable has a name that conforms to SAS naming conventions. Variable names follow the same rules as SAS data set names.

- They can be 1 to 32 characters long.
- They must begin with a letter (A-Z, either uppercase or lowercase) or an underscore (_).
- They can continue with any combination of numbers, letters, or underscores.

Table 2.6 *Variable Name Attributes*

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

VALIDVARNAME= System Option

Note: If you use characters other than the ones that are valid when the VALIDVARNAME system option is set to V7 (letters of the Latin alphabet, numerals, or underscores), then you must express the variable name as a name literal and you must set VALIDVARNAME=ANY. If the name includes either a percent sign (%) or an ampersand (&), then you must use single quotation marks in the name literal in order to avoid interaction with the SAS macro facility.

VALIDVARNAME specifies the rules for valid SAS variable names that can be created and processed during a SAS session.

Syntax, VALIDVARNAME=

VALIDVARNAME= *V7|UPCASE|ANY*

V7 specifies that variable names must follow these rules:

- SAS variable names can be up to 32 characters long.
- The first character must begin with a letter of the Latin alphabet (A - Z, either uppercase or lowercase) or an underscore (_). Subsequent characters can be letters of the Latin alphabet, numerals, or underscores.
- Trailing blanks are ignored. The variable name alignment is left-justified.
- A variable name cannot contain blanks or special characters except for an underscore.
- A variable name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, `cat`, `Cat`, and `CAT` all represent the same variable.
- Do not assign variables the names of special SAS automatic variables (such as _N_ and _ERROR_) or variable list names (such as _NUMERIC_, _CHARACTER_, and _ALL_) to variables.

UPCASE specifies that the variable name follows the same rules as V7, except that the variable name is uppercase, as in earlier versions of SAS.

ANY specifies that SAS variable names must follow these rules:

- The name can begin with or contain any characters, including blanks, national characters, special characters, and multi-byte characters.
- The name can be up to 32 bytes long.
- The name cannot contain any null bytes.
- Leading blanks are preserved, but trailing blanks are ignored.
- The name must contain at least one character. A name with all blanks is not permitted.
- A variable name can contain mixed-case letters. SAS stores and writes the variable name in the same case that is used in the first reference to the variable. However, when SAS processes a variable name, SAS internally converts it to uppercase. Therefore, you cannot use the same variable name with a different combination of uppercase and lowercase letters to represent different variables. For example, `cat`, `Cat`, and `CAT` all represent the same variable.

CAUTION:

> **Throughout SAS, using the name literal syntax with SAS member names that exceed the 32-byte limit or have excessive embedded quotation marks might cause unexpected results.** The VALIDVARNAME=ANY system option enables compatibility with other DBMS variable (column) naming conventions, such as allowing embedded blanks and national characters.

Type

A variable's type is either character or numeric.

- Character variables, such as Name (shown below), can contain any values.
- Numeric variables, such as Total (shown below), can contain only numeric values (the numerals 0 through 9, +, -, ., and E for scientific notation).

Table 2.7 Type Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR8.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Length

A variable's length (the number of bytes used to store it) is related to its type.

- Character variables can be up to 32,767 bytes long. In the example below, Name has a length of 20 characters and uses 20 bytes of storage.
- All numeric variables have a default length of 8 bytes. Numeric values are stored as floating-point numbers in 8 bytes of storage.

Table 2.8 Length Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR9.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Format

Formats are variable attributes that affect how data values are written. Formats do not change the stored value in any way; they merely control how that value is displayed. SAS software offers a variety of character, numeric, and date and time formats. You can also create and store your own formats. To write values out using a particular form, you select the appropriate format.

For example, to display the value `1234` as $1,234.00 in a report, you can use the DOLLAR9.2 format, as shown for Total below.

Table 2.9 Format Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number

Variable	Type	Length	Format	Informat	Label
Total	Num	8	DOLLAR9.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Usually you have to specify the maximum width (*w*) of the value to be written. Depending on the particular format, you might also need to specify the number of decimal places (*d*) to be written. For example, to display the value **`5678`** as 5,678.00 in a report, you can use the COMMA8.2 format, which specifies a width of 8 including 2 decimal places.

TIP You can permanently assign a format to a variable in a SAS data set, or you can temporarily specify a format in a PROC step to determine how the data values appear in output.

Informat

Whereas formats write values out using some particular form, informats read data values in certain forms into standard SAS values. Informats determine how data values are read into a SAS data set. You must use informats to read numeric values that contain letters or other special characters.

For example, the numeric value **`$12,345.00`** contains two special characters, a dollar sign ($) and a comma (,). You can use an informat to read the value while removing the dollar sign and comma, and then store the resulting value as a standard numeric value. For Total below, the COMMA10. informat is specified.

Table 2.10 *Informat Attribute*

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR9.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

Label

A variable can have a label, which consists of descriptive text up to 256 characters long. By default, many reports identify variables by their names. You might want to replace the name with more descriptive information about the variable by assigning a label to the variable.

For example, you can label Policy as Policy Number, Total as Total Balance, and Name as Patient Name to display these labels in reports.

Table 2.11 Label Attribute

Variable	Type	Length	Format	Informat	Label
Policy	Char	8			Policy Number
Total	Num	8	DOLLAR9.2	COMMA10.	Total Balance
Name	Char	20			Patient Name

You can use labels to shorten long variable names in your reports.

SAS Indexes

An index is a separate file that you can create for a SAS data file in order to provide direct access to a specific observation. The index file has the same name as its data file and a member type of INDEX. Indexes can provide faster access to specific observations, particularly when you have a large data set. The purpose of SAS indexes is to optimize WHERE expressions and to facilitate BY-group processing. For more information, see "Specifying WHERE Expressions" on page 72 and see "Group Processing Using the BY Statement" on page 183.

Extended Attributes

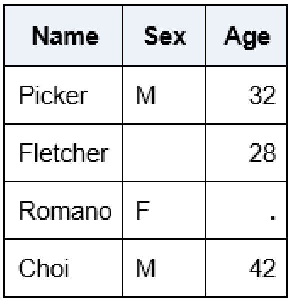
Extended attributes are user-defined metadata that is defined on a data set or on a variable (column). Extended attributes are represented as name-value pairs and are created using the DATASETS procedure.

TIP You can use PROC CONTENTS to display data set and variable extended attributes.

Chapter Quiz

Select the best answer for each question. Check your answers using the answer key in the appendix.

1. How many observations and variables does the data set below contain?



Name	Sex	Age
Picker	M	32
Fletcher		28
Romano	F	.
Choi	M	42

   a. 3 observations, 4 variables

   b. 3 observations, 3 variables

c. 4 observations, 3 variables

d. cannot tell because some values are missing

2. How many program steps are executed when the program below is processed?

```
data user.tables;
   infile jobs;
   input date yyddmm8. name $ job $;
run;
proc sort data=user.tables;
   by name;
run;
proc print data=user.tables;
run;
```

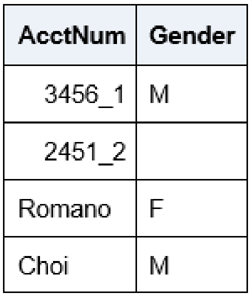
a. three

b. four

c. five

d. six

3. What type of variable is the variable AcctNum in the data set below?



AcctNum	Gender
3456_1	M
2451_2	
Romano	F
Choi	M

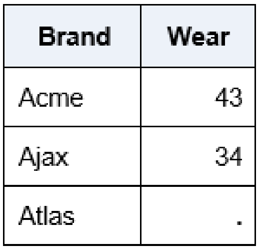
a. numeric

b. character

c. can be either character or numeric

d. cannot tell from the data shown

4. What type of variable is the variable Wear in the data set below?



Brand	Wear
Acme	43
Ajax	34
Atlas	.

a. numeric

b. character

c. can be either character or numeric

d. cannot tell from the data shown

5. With the system option VALIDVARNAME=ANY, which of the following variable names is valid?

a. 4BirthDate

b. $Cost

c. _Items_

d. Tax-Rate

e. All of the above

6. Which of the following files is a permanent SAS file?

a. Sashelp.PrdSale

b. Sasuser.MySales

c. Profits.Quarter1

d. all of the above

7. In a DATA step, how can you reference a temporary SAS data set named Forecast?

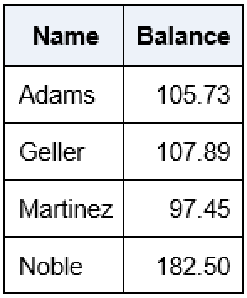
a. Forecast

b. Work.Forecast

c. Sales.Forecast (after assigning the libref Sales)

d. only a and b above

8. What is the default length for the numeric variable Balance?



Name	Balance
Adams	105.73
Geller	107.89
Martinez	97.45
Noble	182.50

a. 5

b. 6

c. 7

d. 8

9. How many statements does the following SAS program contain?

```
proc print data=clinic.admit
                label double;
   var ID Name Sex Age; where Sex=F;
   label Sex='Gender'; run;
```

a. three

b. four

c. five

d. six

10. What is a SAS library?

a. collection of SAS files, such as SAS data sets and catalogs

b. in some operating environments, a physical collection of SAS files

c. a group of SAS files in the same folder or directory

d. all of the above

Index

Special Characters

ALL keyword 127, 385, 387
CHARACTER keyword 385, 386
ERROR automatic variable
  DATA step iterations and 148, 266, 270
  functionality 141
  initializing variables 144
  PUT statement and 160
_N_ automatic variable
  DATA step processing and 147, 266, 269, 272
  functionality 141, 516
  PUT statement and 160
NULL keyword 118
NUMERIC keyword 385, 386
TEMPORARY keyword 396
@ (trailing at-sign) 481, 488, 509
@@ (double trailing at-sign) 481
@n pointer control 403
$BINARYw. informat 406
$VARYINGw. informat 406
$w. format 91
$w. informat
  ampersand modifier 432
  functionality 406, 407
#n pointer control
  combining pointer controls 473
  reading records non-sequentially 469
+n pointer control 405

A

ampersand modifier 431, 434
AND operator
  examples 236
  in logical operations 235
  in SAS expressions 113
  in WHERE statement, PRINT procedure 73
APPEND procedure
  examples 287
  FORCE option 288
  functionality 278, 286
  requirements 287
appending data sets 278, 286
arguments in functions 315
arithmetic operators in SAS expressions 112
array elements
  assigning initial values 394
  creating temporary 396
  referencing 387
  specifying 385
  variables lists as 385
ARRAY statement
  arrays of character variables 393
  assigning initial values to arrays 394
  creating temporary array elements 396
  creating variables 391
  defining one-dimensional arrays 384
  DO loop processing 389, 390
  referencing array elements 387
  specifying array name 385
  specifying dimensions 385
  specifying elements 385
  syntax 384
  variable lists as elements 385
arrays 384
  concatenating names 392
  creating one-dimensional 383
  creating variables 391
  defining 384
  DO loop processing 389, 390
  function arguments and 316
  specifying dimensions 385
  specifying names 385
assignment statements 113
  conditional processing 235
  DATA step processing 146, 268, 271
  date constants 115
  examples 113
  positioning SUBSTR function 345
  SAS expressions in 112
asterisk (*) 191
attributes
  See variable attributes
automatic variables
  DATA step processing 270

PUT statement 160
testing programs 160

B

BEST12. format 321
BESTw. format 180
blanks
compressing 359
browsers 37
preferred web browser 38
BY clause, DO statement 370, 372
BY statement
DESCENDING option 294
group processing with 183, 259
interleaving data sets 278, 289
match-merging data sets 278, 291
PRINT procedure 77, 79
SORT procedure 75, 294
syntax 183
BY variables
finding observations in subgroups 261
interleaving data sets 289

C

calculations
dates and times in 451
case sensitivity
format values 168
CATX function
functionality 349
syntax 349
CEIL function 360
character strings 159
PUT statement 159
searching 350, 351
specifying delimiters 341
testing programs 159
character variables 407
array processing and 386
creating arrays of 392
embedded blanks 431
informat support 407
reading range of 421
removing trailing blanks 347, 348
replacing 344
searching for strings 350
specifying value length 421
character-to-numeric conversions 316, 317
CLASS statement, MEANS procedure 182
cleaning data 155
CLM statistic 178
code editing windows
resubmitting revised programs 54
colon (:) modifier 431, 433
column input
mixing styles 439
nonstandard data values and 402
column pointer controls
@n 403
+n 405
functionality 404
columns
See variables
combining data sets 278
by appending 278, 286
by concatenating 278, 284
by interleaving 278, 289
by match-merging 278, 291
by one-to-one reading 278, 280
excluding unmatched observations 302
methods for 278
renaming variables 300
comma (,) 21, 394
COMMA10. informat 21
COMMA12.2 format 93
COMMA6. format 244
COMMA8.2 format 21
COMMA9. informat 320
COMMAw.d format 91, 92
COMMAw.d informat
example 406
functionality 408, 434
modify list input and 433
common errors
missing RUN statement 52
missing semicolon (;) 52
unbalanced quotation mark 52
comparison operators 235
compilation phase (DATA step)
data set variables 142
descriptor portion of data sets 143
diagnosing errors in 152
input buffer 141
match-merge processing 296
program data vector 141
reading data sets 266
syntax checking 142
COMPRESS function
compressing blanks 359
examples 359
compressing character strings
blanks 359
concatenating array names 392
concatenating data sets 278, 284, 285
concatenation operator (||) 321, 322
conditional processing
assigning variable values 234, 245
assignment statements 235

comparison operators 235
DO groups 248
DO loops 376
iterative DO statement 377
logical operators 235
providing alternative actions 237
PUT statement and 161
SELECT groups 245
single observations per detail records 509
testing programs 161, 238

CONTAINS operator 73

CONTENTS procedure
reading Excel data 122, 127
VARNUM option 35
viewing library contents 31

CONTENTS statement, DATASETS procedure 35

converting data
See data conversion

counter variables 497

CROSSLIST option, TABLES statement (FREQ) 195

CSS statistic
MEANS procedure 178

current record, holding with line-hold specifier 481, 488

CV statistic
MEANS procedure 178

D

data cleaning 155

data conversion 316
character-to-numeric 317
lower case 354
numeric-to-character 316
numeric-to-character conversion 321
upper case 353

data sets
See also combining data sets
See also match-merging data sets
BY-group processing 259
data portion 16
descriptor portion 15, 143
detecting end of 264
dropping and keeping variables 241
iteratively processing data 375
listing 109
manipulating data 257
missing values 17
naming 104
naming conventions 14
observations (rows) 16
reading 255, 266
specifying observations via system options 46
summarized using MEANS procedure 184
summarized using SUMMARY procedure 186
testing programs 159
variable attributes 17, 142
variables and 17
viewing descriptor information 35

DATA step
characteristics 7
checking processing 108
column input 105
compilation phase 141, 266, 296
creating data sets from raw data files 101
creating raw data files 118
creating/modifying variables 111
debugging 152
describing data in programs 106
execution phase 267, 297
functionality 139
invalid data 109
iterations of 148, 490
listing data sets 109
manipulating data 257
naming data sets 104
nonstandard numeric data in 105
PROC IMPORT 121
processing informats 409
reading Excel data 121, 122, 125
reading raw data files 109
repeating blocks of data 482
single observations per detail records 511
single observations per header records 520
specifying raw data files 104
specifying variable names 107
standard numeric data in 105
submitting 108
subsetting data 115
syntax 104
varying number of repeating fields 497
verifying data 108
writing 104, 125

data validation 155

DATALINES statement 116, 117

DATASETS procedure 33, 35

DATDIF function 325, 338

DATE function 325, 332

DATE7. format 93

DATE9. format 93, 327, 328

dates and times
displaying 482

handling four-digit years 44
handling two-digit years 42
in calculations 451
informat support 447
manipulating for output 42
manipulating with functions 323
DATETIMEw. format 447
DATETIMEw. informat 406, 450
DATEw. format
examples 91, 447, 482
functionality 437
DATEw. informat
examples 406
functionality 449
syntax 449
DAY function
functionality 314
manipulating date values 326
syntax 326
typical use 325
debugging
cleaning data 155
diagnosing errors in compilation phase 152
diagnosing errors in execution phase 152
validating data 155
decimal places, limiting 180
DELETE statement
example 240
in IF-THEN statement 240
delimiters
DLM= option, FILE statement 437
DLM= option, INFILE statement 424, 429
DSD option, FILE statement 438
DSD option, INFILE statement 428
for SCAN function 341
free-format data 424, 429
specifying multiple 342
within quoted strings 426
DESCENDING option
BY statement, SORT procedure 75
descriptive statistics 176
creating summarized data sets 184, 186
creating tables in list format 194
group processing with BY statement 183
group processing with CLASS statement 182
limiting decimal places 180
procedure syntax 176
producing frequency tables 187, 191
selecting statistics 177
specifying variables in FREQ procedure 189
specifying variables in MEANS procedure 181
detail record, single observation per
conditional processing 509
DATA step processing 511
displaying results 514
dropping variables 511
overview 508
reading detail records 510
retaining variable values 509
DIM function 390
DLM= option
FILE statement 437
INFILE statement 424, 429
DO groups
indenting 249
iteratively processing data 373
nesting 249
single observations per header records 516
DO loops
conditionally executing 376
constructing 368
counting iterations 371
decrementing 372
functionality 370
index variables 371, 372, 374, 388
INPUT statement example 490
iterative DO statement 377
nesting 373
processing array elements 389, 390
specifying series of items 372
DO statement
See also iterative DO statements
BY clause 370, 372
grouping statements 248
TO clause 369
DO UNTIL statement 249, 376, 377
DO WHILE statement
counter variables 497
executing 497
functionality 249, 376, 377, 496
syntax 377
Document destination 204
dollar sign ($)
arrays of character variables 392
in format names 168
informats and 21, 407
name literals and 124, 125
DOLLAR12.2 format 93
DOLLAR8.2 format 20, 93
DOLLAR9.2 format 93
DOLLARw.d format 91
DOUBLE option, PRINT procedure 82
double spacing LISTING output 82
double trailing at-sign (@@) 481

DROP statement 242, 384
DROP= data set option
  assigning initial values to arrays 395
  determining when to specify 258
  examples 511, 520
  selecting variables 241
DSD option
  FILE statement 438
  INFILE statement 428

E
END statement
  assigning values conditionally 245
  grouping statements 248
end-of-file marker 150
END= option, SET statement 264
engines
  See SAS engines
eq (=) operator 235
error handling 53
  correcting common errors 52
  error types 51
  IF-THEN/ELSE statement 161
  in DATA step compilation phase 152
  in DATA step execution phase 152
  interpreting messages 51
  invalid data 109
  invalid option 61
  resubmitting revised programs 53
  semicolon errors 57
  unbalanced quotation marks 59, 60
  validating or cleaning data 155
Excel data
  See Microsoft Excel data
execution phase (DATA step)
  diagnosing errors in 152
  end of processing actions 146, 151
  end-of-file marker 150
  initializing variables 144
  input data 144
  input pointer 144
  iterations of DATA step 148
  match-merge processing 297
  reading data sets 267
expressions
  See SAS expressions
external files
  See raw data files

F
FILE statement
  DATA step processing 270
  DLM= option 437
  DSD option 438
  functionality 118
  syntax 118
FILENAME statement
  creating data sets from raw data files 101, 102
  defining aggregate storage location 103
  describing data in programs 106
  listing data sets 109
  naming data sets 104
  specifying raw data files 104
  syntax 102
filerefs
  associating with external files 102
  fully qualified filenames in 102, 103
  in aggregate storage locations 103
files
  See raw data files
  See SAS files
FIND function
  examples 352
  functionality 351
  syntax 351
FIRST.variable
  examples 259
  finding observations in subgroups 261
  values supported 259
FIRSTOBS= system option 44
fixed-length records 410
FLOOR function 360
FLOWOVER option, INFILE statement 496
FMTLIB keyword 171
FOOTNOTE statement
  canceling 85
  examples 84
  modifying 85
  quotation marks in 59
  specifying in list reports 82
FOOTNOTES command 83
FOOTNOTES window 83
FORCE option, APPEND procedure 288
FORMAT procedure
  FMTLIB keyword 171
  functionality 165, 166
  invoking 166
  LIBRARY= option 166
  syntax 166
  VALUE statement 167, 168, 170
FORMAT statement
  array name restrictions 384
  assigning formats to variables 170, 243, 457
  displaying date/time values 482
  formatting dates 327, 328
  functionality 90
  syntax 90

formats 20
  assigning permanent 243, 457
  assigning to variables 170, 457
  decimal places 92
  defining unique 167
  examples 93
  field widths 92
  for variables 90
  functionality 90
  permanently assigned 93
  specifying 91
  storing 166
  storing permanently 166
  writing numeric values 180
formatted input
  @n pointer control 403
  +n pointer control 405
  mixing styles 439
  modified list input comparison 435
  reading columns in any order 404
  syntax 403
forward slash (/)
  specifying multiple delimiters 342
forward slash (/) pointer control
  combining pointer controls 473
  reading records sequentially 462
FRAME= option
  syntax 210
free-format data
  See raw data files
FREQ procedure
  See also TABLES statement, FREQ procedure
  detecting invalid data 155
  producing frequency tables 187, 191
  specifying variables 189
  suppressing table information 196
  syntax 155, 187
frequency tables
  creating in list format 194
  n-way 187, 192
  one-way 187
  suppressing information 196
  two-way 191
functions
  arguments 315
  arrays and 315
  character-to-numeric conversions 317
  converting data 316
  manipulating date/time values 323
  syntax 315
  target variables and 316
  variable lists 315

G
ge (>=) operator 235
group processing
  finding observations in subgroups 261
  with BY statement 183, 259
  with CLASS statement 182
gt (>) operator 235

H
header record, single observation per
  DATA step processing 520
  determining end of external file 518
  DO group actions 516
  overview 515
  reading detail records 517
HEXw. informat 406
hierarchical files 507
  creating observation per detail record 508
  creating observation per header record 515
HIGH keyword 168
HTML destination 204
HTML Link and Path Options
  URL= suboption 211
HTML output 36
  creating 36
  ODS overview 204
  overview 206
  viewing 37
HTML Output
  Appearance of HTML 214
  Frame Files 210
  Link and Path Options 211, 212
  Specify Link and Path 211
  Table of Contents 208, 210
HTML Table of Contents
  CONTENTS= option 210
hyphen (-) 342
hypothesis testing 178

I
ID statement, PRINT procedure
  BY statement and 79
  VAR statement and 71
IF-THEN statement
  assigning values conditionally 234
  cleaning data 157
  conditionally executing statements 510
  DELETE statement in 240
  DO groups 248
  ELSE statement 237
  examples 234
  for flagging errors 161

SELECT groups 245
single observations per header records 516, 517
syntax 234
testing programs 158
in operator 235
IN= data set option 302
indenting DO groups 249
INDEX function
functionality 350
syntax 350
index variables in DO loops 371, 372, 374, 388
INFILE statement
column input 105
creating data sets from raw data files 101
DATA step processing 144, 270
describing data in programs 106
DLM= option 424, 429
DSD option 428
FLOWOVER option 496
listing data sets 109
MISSOVER option 427, 496
OBS= option 108, 109
PAD option 412
SCANOVER option 496
single observations per header records 519
specifying raw data files 104
STOPOVER option 496
syntax 104, 519
TRUNCOVER option 496
informats 21
ampersand modifier 432
components 447
DATA step processing 409
modifying list input 435
reading character values 407
reading dates and times 447
reading numeric data 407
specifying 448
initializing variables 144, 233
input buffer 141, 148
INPUT function
character-to-numeric conversion 316, 319
examples 320
syntax 319
INPUT statement
creating data sets from raw data files 101
DATA step processing 144, 148
describing data in programs 106
DO loop example 490
examples 106
line-hold specifier 481, 489
listing data sets 109
mixing input styles 441
pointer controls 403, 405, 462, 469
processing informats 409
processing list input 419
specifying informats 448
syntax 106, 403, 405, 418, 448
variable-length records 411
INT function 360
INTCK function
examples 336
functionality 325, 335
syntax 335
interface library engines 30
interleaving data sets
example 290
functionality 278, 289
Internal Browser option 37
INTNX function 325, 337
invalid data 109, 155
invalid option 61
iterative DO statements 376
conditional clauses 377
conditional executing 376
DATA step example 490
DIM function 390
nesting DO loops 373

J

JULIANw. informat 406

K

KEEP statement 242, 384
KEEP= data set option
determining when to specify 258
selecting variables 241
KURTOSIS statistic, MEANS procedure 178

L

LABEL option, PRINT procedure 88, 244
LABEL statement 243
array name restrictions 384
assigning labels in multiple 89
assigning labels in single 89
example 88
functionality 88
syntax 88
labels
assigning descriptive 88
assigning for variables 21

assigning permanent 93, 243
LAST.variable
examples 259
finding observations in subgroups 261
values supported 259
LCLM statistic, MEANS procedure 178
le (<=) operator 235
leading blanks, removing 349
LEFT function 347
LENGTH statement
ampersand modifier 432
array name restrictions 384
examples 239
functionality 238, 342
specifying character value length 423
length, variable 20, 342
LIBNAME statement
assigning librefs 29
creating data sets from raw data files 101
defining SAS libraries 28
describing data in programs 106
listing data sets 109
naming data sets 104
reading Excel data 121
referencing files in other formats 30
specifying raw data files 104
syntax 28
libraries
See SAS libraries
LIBRARY= option, FORMAT procedure 166
librefs
assigning 28
defined 13
lifespan of 29
verifying 29
line pointer controls
/ (forward slash) 462
#n 469
combining 473
functionality 462
reading records non-sequentially 469
reading records sequentially 462
line-hold specifiers
double trailing at-sign 481
overview 481
trailing at-sign 481, 488, 509
list input
See modified list input
LIST option, TABLES statement (FREQ) 194
list reports
creating 178
creating tables for 194
double spacing output 82
formatting data values 90
generating column totals 76
identifying observations 70
selecting observations 179
selecting variables 69
sorting data 74
specifying footnotes 82
specifying titles 82
LISTING destination 204
log messages 8
Log window
clearing 53
resubmitting revised programs 54
logic errors
PUTLOG statement 55
logical operators
conditional processing 235
in SAS expressions 113
LOW keyword 168
LOWCASE function 351, 354
lt (<) operator 235

M

Markup Languages Family destination 204
match-merging data sets 295
compilation phase 296
examples 292
execution phase 297
functionality 278, 291
handling missing values 298
handling unmatched observations 298
selecting data 292
MAX statistic
MEANS procedure 178, 185
MAXDEC= option, MEANS procedure 180
MDY function
examples 331
functionality 325
missing values 332
MEAN statistic
MEANS procedure 178, 185
MEANS procedure
BY statement 183
CLASS statement 182
creating summarized data sets 184
descriptive statistics 178
detecting invalid data 155, 156
functionality 176
hypothesis testing 178
keywords supported 177
limiting decimal places 180
MAXDEC= option 180
OUTPUT statement 184

quantile statistics 178
selecting statistics 177
specifying variables 181
syntax 156, 176
VAR statement 156, 181
MEDIAN statistic, MEANS procedure 178
MERGE statement
match-merging data sets 278, 291
RENAME= data set option 301
RETAIN statement and 233
syntax 291
merging data sets 289
Microsoft Excel data 130
CONTENTS procedure 122, 127
creating worksheets 130
DATA statement 122, 125
LIBNAME statement 121
name literals 125
named ranges 126
overview 121
PRINT procedure 122, 125, 126
PROC IMPORT 121
referencing workbooks 123
RUN statement 122, 125
SAS/ACCESS LIBNAME statement 121, 122, 129
SET statement 122
steps for reading 122
WHERE statement, DATA step 125
writing the DATA step 125
MIN statistic
MEANS procedure 178, 185
missing values
at beginning/middle of record 428
at end of record 427
in match-merge processing 298
MDY function and 332
overview 17
MISSOVER option, INFILE statement 427, 496
MMDDYY10. format 93
MMDDYY8. format 93
MMDDYYw. format 91
MMDDYYw. informat
examples 406, 447
functionality 448
syntax 448
MODE statistic
MEANS procedure 178
modified list input 418
ampersand modifier 431
colon modifier 431, 433
formatted input comparison 435
mixing styles 439
processing 418
MONTH function
examples 327
functionality 314
manipulating date values 326
syntax 326
typical use 325
multiple observations from single record
overview 479
reading repeating blocks of data 480
reading same number of repeating fields 487
reading varying number of repeating fields 495
multiple records, single observation from
line pointer controls 462, 473
overview 461
reading records non-sequentially 469
reading records sequentially 462

N

N statistic
MEANS procedure 178, 185
n-way frequency tables 187, 192
naming conventions
for variables 18, 107
SAS data sets 14
ne (^=) operator 235
NENGOw. informat 406
nesting
DO groups 249
DO loops 373
NMISS statistic
MEANS procedure 178
NOCOL option, TABLES statement (FREQ) 197
NOCUM option, TABLES statement (FREQ) 191
NOFREQ option, TABLES statement (FREQ) 197
NOOBS option, PRINT procedure 70
NOPERCENT option, TABLES statement (FREQ) 197
NOROW option, TABLES statement (FREQ) 197
NOT operator 235, 236
numeric variables 402
array processing and 386
identifying nonstandard data 402
reading nonstandard data 408
reading range of 420
reading standard data 407
numeric-to-character conversion 316, 321

O

OBS= option, INFILE statement 108, 109
OBS= system option 44
observations 16
  See also combining data sets
  combining from multiple data sets 278
  creating for DO loop iterations 371
  creating from multiple records 462
  creating from single record 480
  creating per detail record 508
  creating per header record 515
  deleting 240
  detecting end of data sets 264
  finding in subgroups 261
  identifying 70
  limiting when testing programs 158
  selecting in list reports 71
  selecting matching 303
  specifying via system options 44
  unmatched 298, 302
  writing explicitly 263
ODS _ALL_ CLOSE statement 205
ODS (Output Delivery System)
  advantages 204
  HTML support 206, 208
  opening and closing destinations 204
ODS destinations 204
ODS EXCEL destination
  TAGATTR= style 223
ODS EXCEL Statement
  Syntax 222
ODS HTML CLOSE statement
  syntax 205
ODS HTML statement
  Table of Contents syntax 208
ODS HTML Statement
  syntax 206
ODS LISTING CLOSE statement 205
ODS PDF Destinations
  Open and Close Statements 216
  Table of Contents 216
ODS PDF Statement
  Statements 216
  syntax 215
ODS RTF
  RTF Formats 221
  RTF Graphics 221
ODS RTF Destinations
  Open and Close Statements 221
ODS RTF Statement
  syntax 220
ODS statements 204
one-to-one reading of data sets
  example 283
  functionality 278, 280, 281
  selecting data 281
one-way frequency tables 187
operands 112
operators
  comparison 235
  concatenation 321, 322
  defined 112
  in SAS expressions 112
  logical 113, 235
OPTIONS statement 41
Options window
  Results tab 38, 40
OR operator
  examples 236
  in logical operations 235
  in SAS expressions 113
  in WHERE statement, PRINT procedure 73
OTHER keyword 168
OTHERWISE statement 245
output
  See also HTML output
  double spacing 82
  manipulating date and times 42
  setting system options 41
Output Delivery System (ODS)
  EXCEL 222, 223
  PDF 215, 216, 219
  RTF 220, 221
Output destination 204
OUTPUT statement
  creating for DO loop iterations 371
  functionality 263
  single observations per header records 519
  syntax 263
OUTPUT statement (MEANS) 185
OUTPUT statement, MEANS procedure
  functionality 184
  syntax 184

P

P1 statistic, MEANS procedure 178
P10 statistic, MEANS procedure 178
P25 statistic, MEANS procedure 178
P5 statistic, MEANS procedure 178
P50 statistic, MEANS procedure 178
P75 statistic, MEANS procedure 178
P90 statistic, MEANS procedure 178
P95 statistic, MEANS procedure 178
P99 statistic, MEANS procedure 178
PAD option, INFILE statement 412
parentheses ()
  for function arguments 315
  logical comparisons in 236
PATH= Option

syntax 212
PDV
See program data vector
PDw.d informat 406
PERCENTw. informat 406
PERCENTw.d informat 406
period (.)
in SAS filenames 13
pointer controls
/ (forward slash) 463
@n 403
#n 469
+n 405
combining 473
functionality 404, 462
reading records non-sequentially 469
reading records sequentially 462
Preferences window
depicted 36, 39, 40
opening 36
Preferred Web Browser option 38
Results tab 36
preferred web browser 38
PRINT option, SUMMARY procedure 186
PRINT procedure
BY statement 77, 79
creating data sets from raw data files 101
creating list reports 68
DOUBLE option 82
ID statement 71, 79
LABEL option 88, 244
listing data sets 109
NOOBS option 70
reading Excel data 122, 125, 126
SUM statement 76, 77
VAR statement 69, 71
WHERE statement 71
Printer Family destination 204
PROBT statistic, MEANS procedure 178
PROC IMPORT
reading Excel data 121
PROC step
characteristics 7
missing RUN statement 57
OPTIONS statement and 41
program data vector 141
DATA step processing 141, 148, 149
match-merge processing 295, 296, 297, 298
reading data sets 266, 268, 272
programming workspace
SAS libraries 11
PROPCASE function 355
PUT function
numeric-to-character conversion 316, 321
syntax 322
PUT statement
automatic variables 160
character strings 159
conditional processing and 161
data set variables 159
functionality 118, 436
syntax 119, 158, 436
testing programs 158, 159, 238
PUTLOG statement 55

Q

Q1 statistic, MEANS procedure 178
Q3 statistic, MEANS procedure 178
QRANGE statistic, MEANS procedure 178
QTR function
functionality 314
manipulating date values 326
syntax 326
typical use 325
quantile statistics 178
quotation marks
common errors 58
delimiters within strings 426
format names and 168
logical operations 236
numeric-to-character conversion 322
reading Excel data 124

R

RANGE statistic
MEANS procedure 178
raw data files 100
See also hierarchical files
creating 118
describing the data 106, 119
formatted input 403
free-format data 417, 436
invalid data 109, 110
mixing input styles 439
modifying list input 431
nonstandard numeric data 402
processing list input 418
reading entire files 109
reading missing values 427
record formats 410
sample record layout 100
specifying 104, 118
specifying character value length 421
records
fixed-length 410

multiple observations from single record 480
single observation from multiple records 462
single observation per detail record 508
single observation per header record 515
variable-length 410
RENAME= data set option 301
renaming variables 300
repeating blocks of data, reading 480
repeating fields
reading same number of 487
reading varying number of 495
Results Viewer window 37
RETAIN statement
DATA step processing 270
initializing sum variables 233
single observations per detail records 509
syntax 233
RIGHT function 347
ROUND function 361
rows
See observations
RTF destination 204
RUN statement
creating data sets from raw data files 101
DATA step processing 148
describing data in programs 106
listing data sets 109
reading Excel data 122, 125

S

SAS engines 30
interface library engines 30
SAS expressions 112
accumulating totals 232
arithmetic operators in 112
logical operators in 113
SELECT statements 247
specifying compound 73
specifying in list reports 72
SAS files 13
naming conventions 14
referencing 13
referencing in other formats 30
storing 12
temporary 14
two-level names 13, 28, 30
SAS formats
See formats
SAS informats
See informats

SAS libraries 11
creating 11
DATASETS procedure 33
defining 11, 28
deleting 12
storing SAS files 12
viewing 31
viewing library contents 31
SAS log 321
SAS programs
DATA step processing 139
error handling 51
log messages 8
processing 8
resubmitting revised 53
results of processing 9
SAS sessions
libref lifespan and 29
OPTIONS statement and 42
SAS statements
See also assignment statements
conditionally executing 509
DO groups 248
executing repeatedly 368
executing when true 496
SELECT groups 245
SAS System Options window
depicted 47
finding options quickly 47
functionality 47
SAS Options Environment pane 47
SAS/ACCESS engines 30
SAS/ACCESS LIBNAME statement
reading Excel data 121, 122
syntax 123
Sashelp library 11
Sasuser library 11
SCAN function
functionality 340
specifying delimiters 341
specifying variable length 342
SUBSTR function versus 346
syntax 340
SCANOVER option, INFILE statement 496
SELECT groups 245
SELECT statement 245
semantic errors 51
semicolon (;)
common errors 57
SET statement
BY statement and 181
concatenating data sets 278, 284
END= option 264
interleaving data sets 278, 289
one-to-one reading 278, 280

reading data sets 255, 266
reading Excel data 122, 125
RETAIN statement and 233
syntax 255
single observation from multiple records
line pointer controls 462, 473
overview 461
reading records non-sequentially 469
reading records sequentially 462
single observation per detail record
conditional processing 509
DATA step processing 511
displaying results 514
dropping variables 511
overview 508
reading detail records 510
retaining variable values 509
single observation per header record
DATA step processing 520
determining end of external file 518
DO group actions 516
overview 515
reading detail records 517
single record, multiple observations from
overview 479
reading repeating blocks of data 480
reading same number of repeating fields 487
reading varying number of repeating fields 495
SKEWNESS statistic, MEANS procedure 178
SORT procedure
BY statement 75, 294
examples 75
sorting data in list reports 74
syntax 74
sorting data in list reports 74
statistics
quantile 178
summary 185
STD statistic
MEANS procedure 178, 185
STDDEV statistic, MEANS procedure 178
STDERR statistic
MEANS procedure 178
STEP statement 125
STOPOVER option, INFILE statement 496
storing
formats 166
SAS files 12, 103
strings
See character strings
STYLE= option
syntax 214, 219
subgroups, finding observations in 261
subsetting data 240
subsetting IF statement
examples 116, 264
finding year 328
functionality 115, 258
selecting matching observations 303
syntax 115
SUBSTR function
functionality 321, 342
positioning 345
replacing text 344
SCAN function versus 346
syntax 343
subtotaling variables 77, 189
sum statement
accumulating totals 232
DO loops and 369
reading data sets 269, 270
syntax 232
SUM statement, PRINT procedure
creating customized layouts 79
generating column totals 76
requesting subtotals 77
syntax 76
SUM statistic
MEANS procedure 178
sum variables, initializing 233
SUMMARY procedure
creating summarized data sets 186
PRINT option 186
summary statistics 185
SUMWGT statistic
MEANS procedure 178
syntax errors 51
system options
changing 47
finding quickly 47
modifying via OPTIONS statement 41
setting 41
specifying observations 44

T

T statistic
MEANS procedure 178
TABLES statement, FREQ procedure 155
creating n-way tables 192
creating two-way tables 191
CROSSLIST option 195
examples 189
LIST option 194
NOCOL option 197
NOCUM option 191
NOFREQ option 197

NOPERCENT option 197
NOROW option 197
syntax 189, 191
target variables
defined 316
missing values and 332
temporary array elements 396
temporary variables 181, 261, 302
testing programs
automatic variables 160
character strings 159
conditional processing 161, 238
data set variables 159
hypothesis testing 178
limiting observations 158
PUT statement 158
TIME function 325
TIMEw. format 447
TIMEw. informat 406, 449
TITLE statement
canceling 85
examples 83
modifying 85
quotation marks in 59
specifying in list reports 82
TITLES command 83
TITLES window 83
TO clause, DO statement 369
TODAY function 325
trailing at-sign (@) 481, 488, 509
trailing blanks, removing 347, 348, 349
TRANWRD function 356
TRIM function 348
TRUNCOVER option, INFILE statement 496
two-way frequency tables 191

U

UCLM statistic, MEANS procedure 178
underscore (_) 18
UNIX environment
SAS library implementation 12, 29
storing files 12
unbalanced quotation marks 59, 60
unmatched observations
excluding 302
handling 298
UPCASE function 351, 353
UPDATE statement 233
USS statistic
MEANS procedure 178

V

validating data 155
VALUE statement, FORMAT procedure
assigning formats to variables 170
functionality 167
HIGH keyword 168
LOW keyword 168
OTHER keyword 168
specifying value ranges 168
syntax 167
VAR statement
MEANS procedure 156, 178, 181, 185
PRINT procedure 69, 71
VAR statistic
MEANS procedure 178
variable attributes 17
data sets 17, 142
format considerations 20
variable lists
as array elements 385
function arguments and 316
variable-length records 410
variables 17
See also arrays
See also character variables
See also numeric variables
See also variable attributes
accumulating totals 232
assigning formats 170, 243, 457
assigning labels 21, 197, 243
assigning values conditionally 234, 245
attributes 17, 20
counter 497
creating or modifying 111
creating/modifying 232
default names 392
DO groups 248
dropping 511
format overview 20
functionality 17
generating totals 76
index 371, 372, 374, 388
informat overview 21
initializing 144, 233
labels for 21
length of 20, 342
missing values 17
naming conventions 18, 107
reading ranges 420
renaming 300
requesting subtotals 77, 189
retaining values 509
SELECT groups 245
selecting in list reports 69
selecting to drop and keep 241
specifying in FREQ procedure 189
specifying in MEANS procedure 181
specifying lengths 238

specifying names in programs 107
subsetting data 240
sum 233
target 316, 332
temporary 181, 261, 302
testing programs 159
types of 19
VARNUM option
CONTENTS procedure 35
CONTENTS statement, DATASETS procedure 35

W

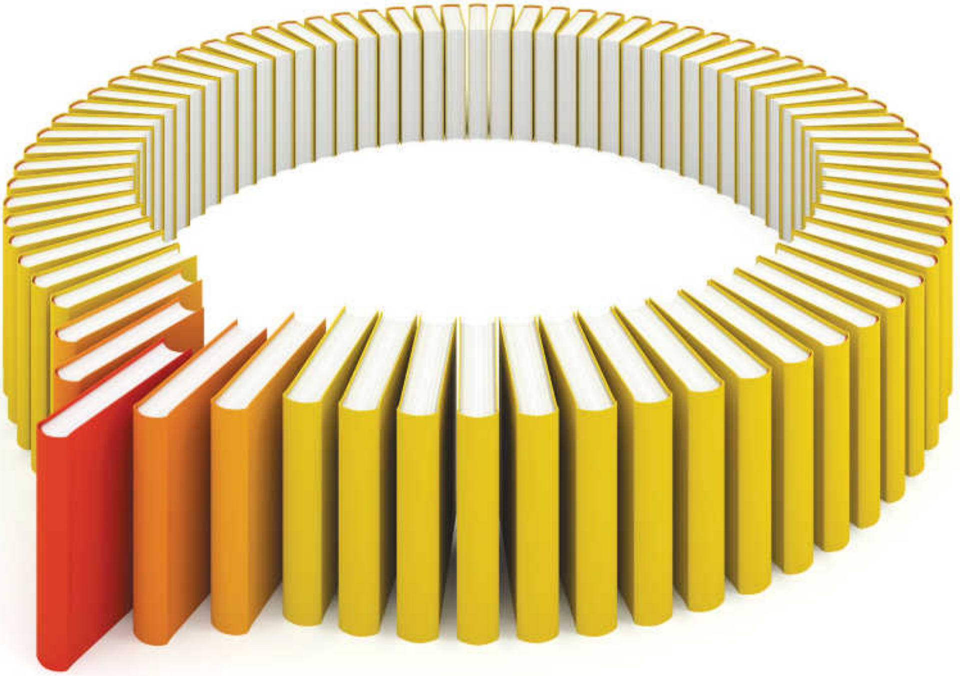
w. format 91
w.d format 91
w.d informat 318, 406, 407
web browsers
See browsers
WEEKDATEw. format 456
WEEKDAY function 325, 329
WHEN statement 245
WHERE statement, DATA step
automatic conversions and 318
reading Excel data 125
WHERE statement, PRINT procedure
CONTAINS operator 73
examples 73
specifying compound expressions 73
specifying expressions 72
syntax 71
Windows environment
SAS library implementation 12, 29
storing files 12
unbalanced quotation marks 59
WORDDATEw. format 457
Work library 11
writing observations explicitly 263

Y

YEAR function
examples 327
functionality 314
manipulating date values 326
syntax 326
typical use 325
YEARCUTOFF= system option
functionality 451
handling four-digit years 44
handling two-digit years 42, 452
YRDIF function 325, 338

Z

z/OS environment
SAS library implementation 12, 29
unbalanced quotation marks 59, 60



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.

SAS THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613