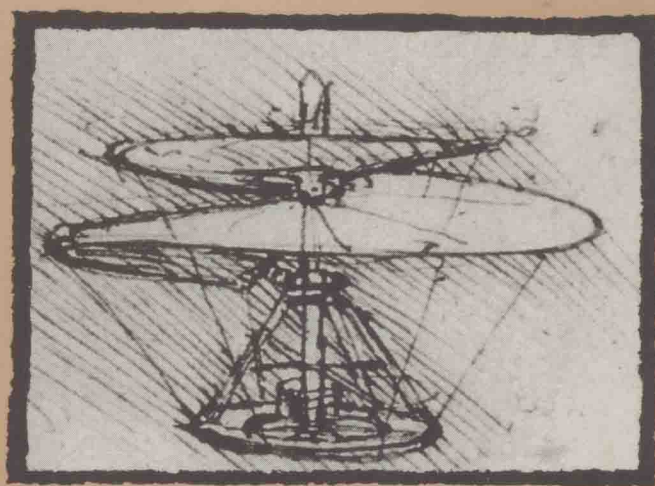


# Applied Statistics and the SAS<sup>®</sup> Programming Language

Fourth Edition



RONALD P. CODY  
JEFFREY K. SMITH

FOURTH EDITION

---

# **Applied Statistics and the SAS Programming Language**

---

***Ronald P. Cody***

ROBERT WOOD JOHNSON MEDICAL SCHOOL

***Jeffrey K. Smith***

RUTGERS UNIVERSITY

PRENTICE HALL, Upper Saddle River, New Jersey 07458

**Library of Congress Cataloging-in-Publication Data**

Cody, Ronald P.

Applied statistics and the SAS programming language / Ronald P.

Cody, Jeffrey K. Smith - 4th ed.

p. cm.

Includes bibliographical references (p. - ) and index.

ISBN 0-13-743642-4 (pbk.)

1. SAS (Computer file) 2. Mathematical statistics--Data

processing. I. Smith, Jeffrey K. II. Title.

QA276.4.C53 1997

519.5'0285'5369--dc21

97-1737

CIP

Acquisition editor: Ann Heath

Editorial assistant: Mindy Ince

Editorial director: Tim Bozik

Editor-in-chief: Jerome Grant

Assistant vice president of production and manufacturing: David W. Riccardi

Editorial/production supervision: Nicholas Romanelli

Managing editor: Linda Mihatov Behrens

Executive managing editor: Kathleen Schiaparelli

Manufacturing buyer: Alan Fischer

Manufacturing manager: Trudy Piscioti

Marketing manager: Melody Marcus

Marketing assistant: Jennifer Pan

Creative director: Paula Maylahn

Cover designer: Jayne Conte



©1997, 1991 by Prentice-Hall, Inc.

Simon & Schuster/A Viacom Company

Upper Saddle River, New Jersey 07458

SAS is a registered trademark of SAS Institute, Inc., Cary, North Carolina

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Cover illustration: From the manuscripts of Leonardo DaVinci; published by Charles Raviasson-Mollien, 6 vols., Paris, 1881-91.

Back cover photo of Ron Cody by Russ Cody.

Printed in the United States of America

10 9 8 7 6 5 4

ISBN: 0-13-743642-4

Prentice-Hall, International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada, Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

---

**Applied Statistics  
and the SAS  
Programming Language**

---

*To our parents,*

**Ralph and Bettie Smith**

**and**

**Philip and Margaret Cody**

---

## Preface to the Fourth Edition

---

When we began creating this fourth edition, several facts were clear: First, SAS software continues to evolve and improve. Second, our programming techniques have also improved. Third, several statistical techniques (such as logistic regression) have become popular and required coverage in this edition.

We have met many readers of earlier editions at meetings and conferences and were delighted to hear good things and constructive criticisms of the book. These we have taken to heart and attempted to improve old material and add relevant new topics. This fourth edition is the result of such reader reaction.

Most researchers are inherently more interested in the substance of their research endeavors than in statistical analyses or computer programming. Yet, conducting such analyses is an integral link in the research chain (all too frequently, the weak link). This condition is particularly true when there is no resource for the applied researcher to refer to for assistance in running computer programs for statistical analyses. *Applied Statistics and the SAS Programming Language* is intended to provide the applied researcher with the capacity to perform statistical analyses with SAS software without wading through pages of technical documentation.

The reader is provided with the necessary SAS statements to run programs for most commonly used statistics, explanations of the computer output, interpretations of results, and examples of how to construct tables and write up results for reports and journal articles. Examples have been selected from business, medicine, education, psychology, and other disciplines.

SAS software is a combination of a statistical package, a data-base management system, and a high-level programming language. Like SPSS, BMDP, Systat, and other statistical packages, SAS software can be used to describe a collection of data and produce a variety of statistical analyses. However, SAS software is much more than just a statistical package. Many companies and educational institutions use SAS software as a high-level data-management system and programming language. It can be used to organize and transform data and to create reports of all kinds. Also, depending on which portions of the SAS system you have installed in your computer (and what type of computer system you are running), you may be using the SAS system for interactive data entry or an on-line system for order entry or retrieval.

This book concentrates on the use of the SAS system for the statistical analysis of data and the programming capabilities of SAS software most often used in educational and research applications.

The SAS system is a collection of products, available from the SAS Institute in Cary, North Carolina. The major products available from the SAS Institute are:

Base SAS ®	The main SAS module, which provides some data manipulation and programming capability and some elementary descriptive statistics
SAS/STAT ®	The SAS product that includes all the statistical programs except the elementary ones supplied with the base package
SAS/GRAPH ®	A package that provides high-quality graphs and maps. Note that “line graphics” (the graphs and charts that are produced by normal character plots) are available in the base and SAS/STAT packages. SAS/GRAPH adds the ability to produce high-quality camera-ready graphs, maps, and charts.
SAS/FSP ®	These initials stand for the Full Screen Product. This package allows you to search, modify, or delete records directly from a SAS data file. It also provides for data entry with sophisticated data checking capabilities. Procedures available with FSP are FSBROWSE, FSEDSIT, FSPRINT, FSLIST, and FSLETTER.
SAS/AF ®	AF stands for the SAS Applications Facility. This product is used by data processing professionals to create “turn key” or menu systems for their users. It is also used to create instructional modules relating to the SAS system.
SAS/ETS ®	The Econometric and Time Series package. This package contains specialized programs for the analysis of time series and econometric data.
SAS/OR ®	A series of operations research programs.
SAS/QC ®	A series of programs for quality control.
SAS/IML ®	The Interactive Matrix Language module. The facilities of IML used to be included in PROC MATRIX in the version 5 releases. This very specialized package allows for convenient matrix manipulation for the advanced statistician.

SAS, SAS/STAT, SAS/GRAPH, SAS/FSP, SAS/AF, SAS/ETS, SAS/OR, SAS/QC, and SAS/IML are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

SAS software now runs on a variety of computers, from personal computers to large multimillion dollar mainframes. The original version of the SAS system was written as a combination of PL/1 and IBM assembly language. Today, SAS software runs under Windows and Windows 95 on IBM compatible minicomputers, on most Macintosh computers, under UNIX on a large number of minicomputers and workstations, on IBM computers under a variety of operation systems, on Digital Equipment VAX computers, and others too numerous to mention. The major reason for the ability of SAS software to run on such a variety of machines is that all SAS software was rewritten in C and designed so that most of the code was system-independent. The conversion of the entire system to the C programming language was one of the largest (and most successful) programming projects ever undertaken. To migrate the SAS system to another computer or operating system, only a small system-dependent portion of code needs to be rewritten. The result is that new versions of SAS software

are made available for all computers very quickly, and the versions of SAS systems from one computer to another are much alike.

Learning to program is a skill that is difficult to teach well. While you will find our examples “reader-friendly” and their logic easy to follow, learning to write your own programs is another matter. Only through lots of practice will your programming skills develop. So, when you finish a chapter, please spend the time doing as many problems as you can. We wish you happy programming.

We express our gratitude to two colleagues who reviewed the manuscript and made many comments beneficial to this revision of the text. Our thanks, therefore, to Sylvia Brown and Robert Hamer, at the Robert Wood Johnson Medical School. Our sincere thanks also to Ann Heath, acquisitions editor for statistics at Prentice Hall, for her encouragement and support, and to Nicholas Romanelli for his superb editing, patience, and good cheer.

RON CODY

JEFFREY SMITH



---

# Contents

---

## ***Applied Statistics and SAS Software***

---

### **Chapter 1 A SAS Tutorial 1**

- A. Introduction 1
- B. Computing with SAS Software: An Illustrative Example 2
- C. Enhancing the Program 7
- D. SAS Procedures 10
- E. Overview of the SAS Data Step 13
- F. Syntax of SAS Procedures 13
- G. Comment Statements 15
- H. References 18

### **Chapter 2 Describing Data 22**

- A. Introduction 22
- B. Describing Data 22
- C. More Descriptive Statistics 26
- D. Descriptive Statistics Broken Down by Subgroups 32
- E. Frequency Distributions 34
- F. Bar Graphs 35
- G. Plotting Data 42
- H. Creating Summary Data Sets with PROC MEANS and PROC UNIVARIATE 45
- I. Outputting Statistics Other Than Means 53
- J. Creating a Summary Data Set Containing a Median 54

### **Chapter 3 Analyzing Categorical Data 58**

- A. Introduction 58
- B. Questionnaire Design and Analysis 59
- C. Adding Variable Labels 63
- D. Adding “Value Labels” (Formats) 66
- E. Recoding Data 70
- F. Using a Format to Recode a Variable 73
- G. Two-way Frequency Tables 75
- H. A Short-cut Way of Requesting Multiple Tables 78

- I. Computing Chi-square from Frequency Counts 79
- J. A Useful Program for Multiple Chi-square Tables 80
- K. McNemar's Test for Paired Data 81
- L. Odds Ratios 83
- M. Relative Risk 86
- N. Chi-square Test for Trend 88
- O. Mantel-Haenszel Chi-square for Stratified Tables and Meta Analysis 90
- P. "Check All That Apply" Questions 92

## **Chapter 4 Working with Date and Longitudinal Data 101**

- A. Introduction 101
- B. Processing Date Variables 101
- C. Longitudinal Data 106
- D. Most Recent (or Last) Visit per Patient 109
- E. Computing Frequencies on Longitudinal Data Sets 110

## **Chapter 5 Correlation and Regression 115**

- A. Introduction 115
- B. Correlation 115
- C. Significance of a Correlation Coefficient 118
- D. How to Interpret a Correlation Coefficient 119
- E. Partial Correlations 120
- F. Linear Regression 121
- G. Partitioning the Total Sum of Squares 124
- H. Plotting the Points on the Regression Line 125
- I. Plotting Residuals and Confidence Limits 126
- J. Adding a Quadratic Term to the Regression Equation 128
- K. Transforming Data 129
- L. Computing Within-subject Slopes 133

## **Chapter 6 T-tests and Nonparametric Comparisons 138**

- A. Introduction 138
- B. T-test: Testing Differences between Two Means 138
- C. Random Assignment of Subjects 141
- D. Two Independent Samples: Distribution Free Tests 143
- E. One-tailed versus Two-tailed Tests 145
- F. Paired T-tests (Related Samples) 146

## **Chapter 7 Analysis of Variance 150**

- A. Introduction 150
- B. One-way Analysis of Variance 150

- C. Computing Contrasts 158
- D. Analysis of Variance: Two Independent Variables 159
- E. Interpreting Significant Interactions 163
- F. N-way Factorial Designs 170
- G. Unbalanced Designs: PROC GLM 171
- H. Analysis of Covariance 174

## **Chapter 8 Repeated Measures Designs 181**

- A. Introduction 181
- B. One-factor Experiments 182
- C. Using the REPEATED Statement of PROC ANOVA 168
- D. Two-factor Experiments with a Repeated Measure on One Factor 189
- E. Two-factor Experiments with Repeated Measures on Both Factors 197
- F. Three-factor Experiments with a Repeated Measure on the Last Factor 202
- G. Three-factor Experiments with Repeated Measures on Two Factors 209

## **Chapter 9 Multiple-Regression Analysis 221**

- A. Introduction 221
- B. Designed Regression 226
- C. Nonexperimental Regression 226
- D. Stepwise and Other Variable Selection Methods 228
- E. Creating and Using Dummy Variables 234
- F. Logistic Regression 235

## **Chapter 10 Factor Analysis 250**

- A. Introduction 250
- B. Types of Factor Analysis 250
- C. Principal Components Analysis 251
- D. Oblique Rotations 258
- E. Using Communalities Other Than One 259
- F. How to Reverse Item Scores 262

## **Chapter 11 Psychometrics 265**

- A. Introduction 265
- B. Using SAS Software to Score a Test 265
- C. Generalizing the Program for a Variable Number of Questions 268
- D. Creating a Better Looking Table Using PROC TABULATE 270
- E. A Complete Test Scoring and Item Analysis Program 273
- F. Test Reliability 276
- G. Interrater Reliability 277

## ***SAS Programming***

### **Chapter 12 The SAS INPUT Statement 280**

- A. Introduction 280
- B. List Directed Input: Data values separated by spaces 280
- C. Reading Comma-delimited Data 281
- D. Using INFORMATS with List Directed Data 282
- E. Column Input 283
- F. Pointers and Informats 284
- G. Reading More than One Line per Subject 285
- H. Changing the Order and Reading a Column More Than Once 286
- I. Informat Lists 286
- J. “Holding the Line”—Single- and Double-trailing @’s 287
- K. Suppressing the Error Messages for Invalid Data 288
- L. Reading “Unstructured” Data 289

### **Chapter 13 External Files: Reading and Writing Raw and System Files 298**

- A. Introduction 298
- B. Data in the Program Itself 298
- C. Reading ASCII Data from an External File 300
- D. INFILE Options 302
- E. Writing ASCII or Raw Data to an External File 304
- F. Creating a Permanent SAS Data Set 305
- G. Reading Permanent SAS Data Sets 307
- H. How to Determine the Contents of a SAS Data Set 308
- I. Permanent SAS Data Sets with Formats 309
- J. Working with Large Data Sets 311

### **Chapter 14 Data Set Subsetting, Concatenating, Merging, and Updating 319**

- A. Introduction 319
- B. Subsetting 319
- C. Combining Similar Data from Multiple SAS Data Sets 321
- D. Combining Different Data from Multiple SAS Data Sets 321
- E. “Table Look up” 324
- F. Updating a Master Data Set from an Update Data Set 326

### **Chapter 15 Working with Arrays 329**

- A. Introduction 329
- B. Substituting One Value for Another for a Series of Variables 329
- C. Extending Example 1 to Convert All Numeric Values of 999 to Missing 331
- D. Converting the Value of N/A (Not Applicable) to a Character Missing Value 332

- E. Converting Heights and Weights from English to Metric Units 333
- F. Temporary Arrays 334
- G. Using a Temporary Array to Score a Test 336
- H. Specifying Array Bounds 338
- I. Temporary Arrays and Array Bounds 338
- J. Implicitly Subscripted Arrays 339

## **Chapter 16 Restructuring SAS Data Sets Using Arrays 343**

- A. Introduction 343
- B. Creating a New Data Set with Several Observations per Subject from a Data Set with One Observation per Subject 343
- C. Another Example of Creating Multiple Observations from a Single Observation 345
- D. Going from One Observation per Subject to Many Observations per Subject Using Multi-dimensional Arrays 347
- E. Creating a Data Set with One Observation per Subject from a Data Set with Multiple Observations per Subject 348
- F. Creating a Data Set with One Observation per Subject from a Data Set with Multiple Observations per Subject Using a Multi-dimensional Array 350

## **Chapter 17 A Review of SAS Functions:**

### **Part I. Functions other than character functions 353**

- A. Introduction 353
- B. Arithmetic and Mathematical Functions 353
- C. Random Number Functions 355
- D. Time and Date Functions 356
- E. The INPUT and PUT Functions: Converting Numerics to Character, and Character to Numeric Variables 358
- F. The LAG and DIF Functions 360

## **Chapter 18 A Review of SAS Functions:**

### **Part II. Character Functions 364**

- A. Introduction 364
- B. How Lengths of Character Variables Are Set in a SAS Data Step 364
- C. Working with Blanks 367
- D. How to Remove Characters from a String 368
- E. Character Data Verification 368
- F. Substring Example 369
- G. Using the SUBSTR Function on the Left-hand Side of the Equal Sign 370
- H. Doing the Previous Example Another Way 371
- I. Unpacking a String 372
- J. Parsing a String 373
- K. Locating the Position of One String within Another String 373

- L. Changing Lower Case to Upper Case and Vice Versa 374
- M. Substituting One Character for Another 375
- N. Substituting One Word for Another in a String 376
- O. Concatenating (Joining) Strings 377
- P. Soundex Conversion 378

## **Chapter 19 Selected Programming Examples 382**

- A. Introduction 382
- B. Expressing Data Values as a Percentage of the Grand Mean 382
- C. Expressing a Value as a Percentage of a Group Mean 384
- D. Plotting Means with Error Bars 385
- E. Using a Macro Variable to Save Coding Time 386
- F. Computing Relative Frequencies 387
- G. Computing Combined Frequencies on Different Variables 389
- H. Computing a Moving Average 391
- I. Sorting within an Observation 392
- J. Computing Coefficient Alphas (or KR-20) in a Data Step 393

## **Chapter 20 Syntax Examples 395**

- |                      |                        |
|----------------------|------------------------|
| A. Introduction 395  | L. PROC LOGISTIC 400   |
| B. PROC ANOVA 396    | M. PROC MEANS 400      |
| C. PROC APPEND 396   | N. PROC NPARIWAY 401   |
| D. PROC CHART 396    | O. PROC PLOT 401       |
| E. PROC CONTENTS 397 | P. PROC PRINT 401      |
| F. PROC CORR 397     | Q. PROC RANK 402       |
| G. PROC DATASETS 397 | R. PROC REG 402        |
| H. PROC FACTOR 398   | S. PROC SORT 403       |
| I. PROC FORMAT 398   | T. PROC TTEST 403      |
| J. PROC FREQ 399     | U. PROC UNIVARIATE 403 |
| K. PROC GLM 399      |                        |

## **Problem Solutions 404**

## **Index 439**

## **A SAS Tutorial**

---

- A.** Introduction
- B.** Computing With SAS Software: An Illustrative Example
- C.** Enhancing the Program
- D.** SAS Procedures
- E.** Overview of the SAS DATA Step
- F.** Syntax of SAS Procedures
- G.** Comment Statements
- H.** References

### ***A. Introduction***

For the novice, engaging in statistical analysis of data can seem as appealing as going to the dentist. If that pretty much describes your situation, perhaps you can take comfort in the fact that this is the fourth edition of this book—meaning that the first three editions sold pretty well, and this time we may get it right. Our purpose for this tutorial is to get you started using SAS software. The key objective is to get one program to run successfully. If you can do that, you can branch out a little bit at a time. Your expertise will grow.

The SAS System is a combination of programs originally designed to perform statistical analysis of data. Other programs you may have heard of are SPSS, BMDP, or SYSTAT. If you look at personal computer magazines, you might run across other programs, primarily designed to run on personal computers. Since its inception, the SAS system has grown to where it can perform a fairly impressive array of nonstatistical functions. We'll get into a little of that in later chapters. For now, we want to learn the most basic rudiments of the SAS system. If you skipped over it, the Preface to the fourth edition contains some history of SAS software development and a more complete overview of the capabilities of SAS software.

To begin, SAS software runs on a wide variety of computers and operating systems (computer people call these platforms), and we don't know which one you have. You may have an IBM compatible personal computer running Windows or, perhaps, Windows 95. You may have a Macintosh computer, or you may be connected to a network or a mainframe computer by way of a modem or network

connection. You may only have a sophisticated VCR, which you think is a computer. If you are unsure of what platform you are using or what version of SAS software you are using, ask someone. As a matter of fact, right now would be a good time to invite the best computer person you know to lunch. Have that person arrive at your office about an hour before lunch so you can go over some basic elements of your system. You need to find out what is necessary on your computer to get the SAS system running. What we can teach you here is how to use the SAS system, and how to adapt to your computer system.

If you are running on a mainframe, you may well be submitting what are called “batch” jobs. When you run batch jobs, you send your program (across a phone line or a network from your personal computer or terminal) to the computer. The computer runs your program and holds it until you ask for it or prints out the results on a high-speed printer. You may even need to learn some Job Control Language (which you have to get from your local computer folks), and then you can proceed.

If you are running on a personal computer, or running in what is called interactive mode on a minicomputer or mainframe, then you need to learn how to use the SAS Display Manager. The look and feel of SAS once you are in the Display Manager is pretty much the same whatever platform you are using. If you are undaunted, take a deep breath and plunge into the real content in the next section. If you are already daunted, take a minute and get that lunch scheduled, then come back to this.

## ***B. Computing with SAS Software: An Illustrative Example***

SAS programs communicate with the computer by SAS “statements.” There are several kinds of SAS statements, but they share a common feature—they end in a semicolon. A semicolon in a SAS program is like a period in English. Probably the most common error found in SAS programs is the omission of the semicolon. This omission causes the computer to read two statements as a run-on statement and invariably fouls things up.

SAS programs are comprised of SAS statements. Some of these statements provide information to the system, such as how many lines to print on a page and what title to print at the top of the page. Other statements act together to create SAS data sets, while other SAS statements act together to run predefined statistical or other routines. Groups of SAS statements that define your data and create a SAS data set are called a DATA step; SAS statements that request predefined routines are called a PROC (pronounced “prock”) step. DATA steps tell SAS programs about your data. They are used to indicate where the variables are on data lines, what you want to call the variables, how to create new variables from existing variables, and several other functions we mention later. PROC (short for PROCEDURE) steps indicate what kind of statistical analyses to perform and provide specifications for those analyses. Let’s look at an example. Consider this simple data set:



SUBJECT NUMBER	GENDER (M or F)	EXAM 1	EXAM 2	HOMEWORK GRADE
10	M	80	84	A
7	M	85	89	A
4	F	90	86	B
20	M	82	85	B
25	F	94	94	A
14	F	88	84	C

We have five variables (SUBJECT NUMBER, GENDER, EXAM 1, EXAM 2, and HOMEWORK GRADE) collected for each of six subjects. The unit of analysis, people for this example, is called an observation in SAS terminology. SAS software uses the term “variable” to represent each piece of information we collect for each observation. Before we can write our SAS program, we need to assign a variable name to each variable. We do this so that we can distinguish one variable from another when doing computations or requesting statistics. SAS variable names must conform to a few simple rules: They must start with a letter, be not more than eight characters (letters or numerals) in length, and cannot contain blanks or certain special characters such as commas, semicolons, etc. (The underscore character (\_) is a valid character for SAS variable names and can be used to make variable names more readable.) Therefore, our column headings of “SUBJECT NUMBER,” or “EXAM 1” are not valid SAS variable names. Logical SAS variable names for this collection of data would be:

SUBJECT      GENDER      EXAM1      EXAM2      HWGRADE

It is prudent to pick variable names that help you remember which name goes with which variable. We could have named our five variables VAR1, VAR2, VAR3, VAR4, and VAR5, but we would then have to remember that VAR1 stands for “SUBJECT NUMBER,” and so forth.

To begin, let’s say we are interested only in getting the class means for the two exams. In reality it’s hardly worth using a computer to add up six numbers, but it does provide a nice example. In order to do this, we could write the following SAS program:

```
DATA TEST; ①
  INPUT SUBJECT 1-2 GENDER $ 4 EXAM1 6-8 EXAM2 10-12 ②
  HWGRADE $ 14;
DATALINES; ③
10 M 80 84 A
7 M 85 89 A
4 F 90 86 B
20 M 82 85 B
25 F 94 94 A
14 F 88 84 C
;
PROC MEANS DATA=TEST; ④
RUN; ⑤
```