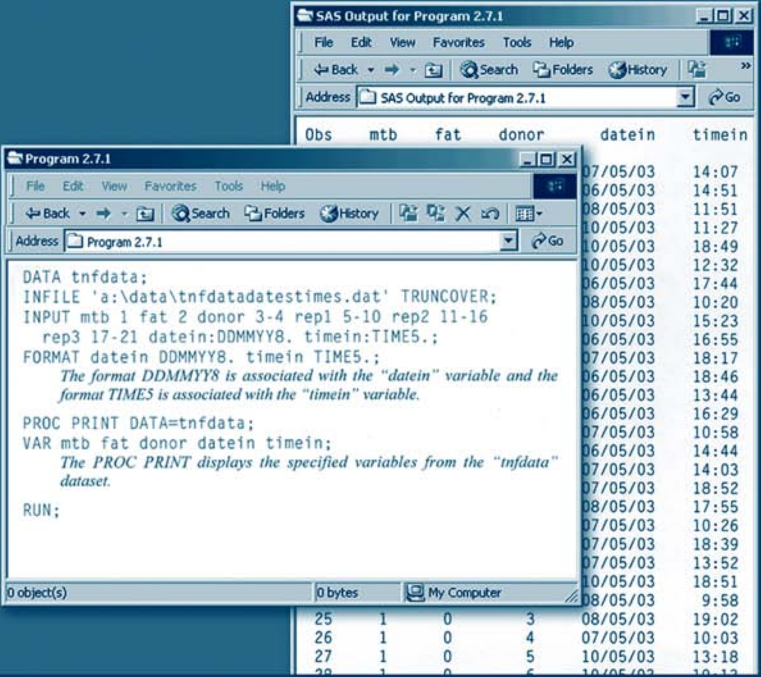
SAS Programming

The One-Day Course

NEIL H. SPENCER


SAS Output for Program 2.7.1
File Edit View Favorites Tools Help
Back Search Folders History
Address SAS Output for Program 2.7.1
Go
Program 2.7.1
File Edit View Favorites Tools Help
Back Search Folders History
Address Program 2.7.1
Go
0 object(s)
0 bytes
My Computer

```
DATA tnfdata;
INFILE 'a:\data\tnfdatadatestimes.dat' TRUNCOVER;
INPUT mtb 1 fat 2 donor 3-4 rep1 5-10 rep2 11-16
  rep3 17-21 datein:DDMMYY8. timein:TIME5.;
FORMAT datein DDMMYY8. timein TIME5.;
```

The format DDMMYY8 is associated with the "datein" variable and the format TIME5 is associated with the "timein" variable.

```
PROC PRINT DATA=tnfdata;
VAR mtb fat donor datein timein;
```

The PROC PRINT displays the specified variables from the "tnfdata" dataset.

```
RUN;
```

```
Obs    mtb    fat    donor      datein     timein

                               7/05/03     14:07
                               6/05/03     14:51
                               8/05/03     11:51
                               0/05/03     11:27
                               0/05/03     18:49
                               0/05/03     12:32
                               6/05/03     17:44
                               8/05/03     10:20
                               0/05/03     15:23
                               6/05/03     16:55
                               7/05/03     18:17
                               6/05/03     18:46
                               6/05/03     13:44
                               6/05/03     16:29
                               7/05/03     10:58
                               6/05/03     14:44
                               7/05/03     14:03
                               7/05/03     18:52
                               8/05/03     17:55
                               7/05/03     10:26
                               7/05/03     18:39
                               7/05/03     13:52
                               0/05/03     18:51
                               8/05/03      9:58
 25      1      0       3    08/05/03     19:02
 26      1      0       4    07/05/03     10:03
 27      1      0       5    10/05/03     13:18
```

CHAPMAN & HALL/CRC

SAS Programming

The One-Day Course

SAS Programming

The One-Day Course

NEIL SPENCER

CHAPMAN & HALL/CRC

A CRC Press Company

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Spencer, Neil.
SAS programming : the one-day course / Neil Spencer
p. cm.
Includes index.
ISBN 1-58488-409-6
1. SAS (Computer file) 2. Statistics—Data processing. I. Title.

QA276.4.S6455 2003
519.5′.0285—dc21 2003055157
CIP

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the authors and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

No claim to original U.S. Government works
International Standard Book Number 1-58488-409-6
Library of Congress Card Number 2003055157
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Dedication

for

Catherine, Laura, Julia, and Helen

Preface

This book has been created by the author as material for a one-day course in SAS programming. The fact that it was developed for a training course is reflected in the concise nature of the presentation. It is not intended to be a general reference book for programming in SAS but is aimed at researchers and students who want to learn the basics of data management, summarising data, and graphics. Once readers have mastered the topics covered in this book, they will be well placed to learn further aspects of SAS programming for themselves. Information on these further aspects can be obtained from the SAS help facilities, relevant manuals, and the SAS Institute Web pages (http://www.sas.com).

Each chapter apart from Chapter 1 contains tasks so that the reader can practice the programming presented in that chapter. The data files needed to carry out the tasks can be found online via: http://www.crcpress.com/e_products/downloads/. The suggested solution programs shown in Appendix C can also be found in electronic form on this website. Also contained at this location are electronic versions of the programs and macros used in the text and the data they access.

The reader is, of course, not obliged to work through the book in just one day, but the material is arranged in such a way that programs and tasks sometimes use aspects of programming or datasets created in previous chapters. It is thus suggested that the reader tackle the topics in the order presented.

SAS Version 8.2 was used by the author in preparing this book, and at the time of writing, SAS was rolling out Version 9. As one would expect, literature describing Version 9 states that programs written for earlier versions of SAS will work under the new version.

Dr. Neil H. Spencer
Principal Lecturer in Statistics, University of Hertfordshire
N.H.Spencer@herts.ac.uk

Author

Neil Spencer is a principal lecturer in statistics at the University of Hertfordshire and head of its Statistical Services and Consultancy Unit. He undertook his undergraduate studies at the University of Reading (where he first encountered SAS), followed by postgraduate study at the University of Southampton and doctoral study at the University of Lancaster. At the University of Hertfordshire, he teaches various courses, including SAS programming to undergraduate students and as a short course through the University's Statistical Services and Consultancy Unit. His research is primarily based in the social sciences and includes multilevel modelling of educational data, statistical computing, and multivariate statistics.

Table of Contents

1 Introducing SAS

SAS is a powerful computer package used for the analysis of data. It is made up of various modules that contain PROCEDURES to undertake specific tasks. The BASE module contains the basics of the SAS system, including data entry and manipulation and simple statistical operations. Purchasers of SAS can then decide what additional modules they require. Examples include the SAS/GRAPH module for graphics, the SAS/IML module (*I*nteractive *M*atrix *L*anguage), and the SAS/STAT module for more advanced statistical operations. There is also a SAS/ASSIST module that allows the user to interact with the system directly, in a similar manner as to many other statistical software packages. It does not, however, use Microsoft® Windows® dialogue boxes that users might expect, and it is much less user friendly than other packages in this regard.

Unless SAS/ASSIST or some other interactive system is being used, the package is operated by the user writing a program and then submitting it to the system. The system then processes the statements and produces output, as appropriate, and a log file, giving details of what operations were carried out.

So why does SAS not have a user-friendly interface? Why does it expect its users to learn how to write programs? The answers lie in the nature of research projects: they are typically undertaken by a group of researchers and are ongoing for a significant amount of time. This means that a record must be kept of exactly what data manipulations, analyses, etc., were accomplished. Keeping programs of statements, which can contain comments on what the programs do, what revisions were made, dates on which the programs were created, and when revisions were made, is an effective tool. These programs can also contain statements that carry out many analyses and data manipulations within the same program. When revisions are made to these more complex programs, the resulting program can be simply submitted to the system, and all the analyses and data manipulations can be carried out together. To do the same using dialogue boxes may necessitate the use of many boxes and take a significant amount of operator time.

SAS is not unique in offering this ability to write programs in a command language. Many packages that use dialogue boxes simply convert the dialogue box information into their own command language for processing. Some keep this conversion hidden from the user, some show it explicitly, and others offer the option of displaying the converted commands.

The capabilities of the SAS system are immense. With this book, the author does not expect readers to become expert programmers in one day but does aim to develop skills that will enable readers to use SAS to a certain level and, perhaps more importantly, put them in a position to further develop programming skills in the particular areas of application that are relevant to their work.

In the programs in this book, words that are part of the SAS programming language are given in uppercase, and words that are defined by the user (e.g., variable names and dataset names) are given in lowercase. SAS, however, is not case sensitive, and the distinction shown here is for clarity only.

1.1 OPERATING SAS

How the user interacts with SAS depends to a large extent on the operating system that is being used. It is not the intention of this book to document in detail the vagaries of using SAS with every possible operating system — its aim is to introduce the programming language. However, some features are common to all systems (in one form or another) and are discussed below. Throughout the rest of the book, efforts are made to avoid detail specific to any particular operating system. The only exception is when the author has judged that mention of features found in the widespread Microsoft Windows operating system is worthwhile (e.g., Section 1.2). When external files are accessed in programs, MS-DOS paths are used.

1.1.1 Writing and Submitting a SAS Program

A SAS program is written in plain text, and any text editor or word processor that can save in plain text format can be used. Once complete, the program is submitted to the SAS software for processing.

For the sake of efficiency, it is a good idea to restrict individual programs to performing just a few tasks. Programs that undertake many different analyses and data manipulations can become unwieldy and difficult to understand.

1.1.2 SAS Log

Once SAS has processed a submitted program, it displays the operations it carried out in a log file, which displays the submitted program in sections with "Notes," "Warnings," and "Errors." Notes are good, warnings are potentially worrisome, and errors tell the user why the program failed. As with any computer program, it is unusual for a SAS program to be written correctly the first time. An understanding of what the log window is telling the user about the program submitted is vital in debugging and is developed as a programmer gains experience.

1.1.3 SAS Output

If the submitted program creates any nongraphical viewable output (i.e., analysis results or tables), then it will be displayed by default in an output file. Graphical output will be produced in a format appropriate to the operating system being used. If the Output Delivery System (discussed in Section 8.1) is used, other files may result.

1.2 OPERATING SAS WITH MICROSOFT WINDOWS

In Version 8 of the Microsoft Windows version of SAS, there is an "Enhanced Program Editor" window, which is where the SAS program resides. When this window is active (has a coloured bar at the top rather than a grey bar), the menu bar at the top of the screen is particular to the Program Editor, and the toolbar buttons will also operate in the window. There are facilities for opening, closing, and saving programs as well as cutting, copying, and pasting. It operates simply as a text editor. An advantage of using the SAS Enhanced Program Editor over other plain text editors is that it automatically colour codes different parts of the program and displays horizontal lines dividing separate parts of the program. This is a valuable aid when writing a program, as when errors in programming syntax are made, the user does not see the expected colour patterns and thus knows that something is wrong. On the toolbar, there is a button that is selected to run the program. The program can also be run via the menu bar: “Run”, “Submit”.

1.2.1 Other Windows

The “Output” window is where the results of the program are shown by default. There is also an accompanying “Results” window that lists the output produced in folder-like forms. SAS organises its output into “pages”. The number of lines that can be fitted onto a page and the maximum length of these lines can be customised by using “Tools”, “Options”, “System…” and then, within the “Log and procedure output control” folder, selecting “SAS log and procedure output”. The option that the user wishes to change is then selected. When the Output window is printed directly from SAS, each new page in SAS requires a fresh sheet of paper. It is frequently the case that a page of SAS output contains only a few lines.

The “Log” window is important when there are errors in the program that was written. It displays the submitted program in sections with colour-coded “Notes”, “Warnings”, and “Errors”.

If graphs are produced by SAS as a result of a program running, they open in separate windows.

1.2.2 Clearing Text from Windows

When an SAS program is run, edited, and rerun successively until a correctly working program is obtained, a lot of redundant text will accumulate in the Log window and potentially in the Output window.

Thus, it is often a good idea to clear all the text from a window before submitting the program. The text that then appears in the Log window must be relevant to the program just submitted, and debugging can take place if necessary. It can also be confusing to have irrelevant text in the Output window, so once a correctly operating program is obtained, it is a good idea to run it again after clearing all previous text from the Output window.