United States General Accounting Office

GAO

Office of Policy

August 1992

Planning, Preparing, Documenting, and Referencing SAS Products

GAO/IMTEC-11.1.2

Preface

The SAS system is a software system for data analysis.[1] It is primarily used for information storage and retrieval, data modification and programming, report writing, statistical analysis, graphics, and file handling. It is used extensively within the General Accounting Office (GAO) to retrieve, analyze, and present data. Evaluators, programmers, and analysts use SAS because it is comprehensive and flexible.

Because properly planning, preparing, documenting, and referencing SAS products can be intricate and demanding, this guide was developed to enable GAO's SAS users to develop products that conform to GAO quality control and workpaper documentation standards. The guide recommends SAS features that are consistent with GAO standards and warns against using those that are not. Although SAS has nonaudit applications, such as management information systems, this guide is intended for audit and program evaluation applications. While this guide is specific to SAS, the principles discussed here apply to other analytical packages.

This guide is targeted toward GAO evaluators and analysts who use SAS or supervise those who do. It complements SAS training and reference manuals and supplements GAO policy directives. Because this guide assumes that the reader understands the syntax and style of SAS statements and procedures, you should consult your technical assistance group if you are unfamiliar with SAS. The Training Institute and other educational centers offer basic courses in using SAS.

This document can also provide guidance to the generalist evaluator by recommending practices in the

[1] SAS is a registered trademark of the SAS Institute Inc., Cary, N.C.

planning, preparing, documenting, and referencing of SAS products.

A team of experienced SAS users, representing both regional and headquarters staff, developed this guide. Major contributors were Arthur D. Foreman and RoJeanne W.L. Liu. If you have any suggestions on improving this guide or need further assistance, please call Mr. Foreman at (513) 684-7120.

Ralph V. Carlone

Ralph V. Carlone
Assistant Comptroller General
Information Management and Technology Division

Werner Grosshans
Assistant Comptroller
General for Policy

Contents

Introduction

Although SAS is flexible, multipurpose, and easy to use, it can easily be misapplied, the results misinterpreted, and errors left undetected. This guide stresses the importance of

- knowing the underlying statistical principles and how to interpret results,
- understanding the structure and characteristics of the data you will be using,
- being familiar with SAS procedures and appropriately applying SAS options and modifiers, and
- being careful to specify the correct parameters for SAS procedures.

Flexibility

SAS software works in a variety of user interfaces and execution methods—batch, interactively with line numbers, interactively with display manager windows, and interactively with menus. Each of these interfaces and methods presents distinct challenges to build quality control into SAS products. Certain interfaces and methods, although convenient and easy to use, are difficult to document. This guide provides hints on when each mode is appropriate and how to use it.

Suitability

This guide suggests what SAS procedures are appropriate for an assignment and when other software would be more effective than SAS. SAS was developed to give analysts one software system to meet a wide variety of computing needs. In addition to providing data retrieval, modification, report writing, and statistical analysis procedures, SAS software provides graphics, operations research, forecasting, and business analysis tools. SAS also provides interfaces to popular data bases, menu-driven facilities for data manipulation, facilities for interactive applications, and a matrix programming language.

Ease of Use

Because of SAS syntax and user interfaces, SAS procedures are easy to use. One method of using SAS is to simply check blocks on a menu. However, appropriately using many SAS procedures requires a technical knowledge of the areas addressed by those procedures. Many procedures require an extensive knowledge of statistics, operations research, and econometrics. In addition, some of the user interfaces do not provide proper GAO workpaper documentation.

Order of Topics

The sequence of topics in this guide follows the normal order of assignment tasks:

- planning work that may involve SAS;
- ensuring correctness in SAS work;
- entering data into SAS from raw data formats, SAS data files, other software formats, and data bases;
- transferring SAS data between computers;
- documenting SAS work;
- referencing SAS work; and
- storing SAS workpapers and files.

Chapter 2

Planning

Using SAS to support an audit assignment requires careful planning. To correctly decide if you should use SAS and, if so, how, you must have explicitly defined the assignment's objectives and the specific questions to be answered. Planning to use SAS involves

- defining the analysis techniques you will use,
- determining the source of data,
- estimating staff proficiency in SAS and statistics,
- determining available computer resources, and
- determining available support material.

The result of this planning effort is an initial SAS work plan. This plan, developed after defining audit objectives and specific information needs, should be included as part of the overall audit plan or evaluation design.

Deciding to Use SAS

The GAO Project Manual describes significant characteristics of various computer software packages available to GAO staff. Selecting the most appropriate package(s) depends on matching the assignment's needs with a software's strengths. The following questions will help you decide if SAS is the most appropriate package.

What Audit Techniques Do You Need?

SAS works well for data retrieval, statistical analysis, operations research, econometrics, and quality control problems and may be appropriate when the assignment calls for these techniques. For example, if the assignment calls for a statistical analysis, SAS is a top candidate. However, if simulation models are a significant portion of the assignment, SAS would not be the best choice because of the extensive programming effort required. Also, although SAS has extensive analytical graphic capabilities, SAS graphics do not meet GAO's visual communications standards. Therefore, do not use SAS graphics for presentation.

Instead, use GAO packages such as Instant Chart or Text Frame.

SAS may be used in conjunction with other products. For example, if your assignment calls for creating a relational data base from which selected items need complex statistical analysis, you could select a data base package, such as dBASE, to gather and structure the data and then use SAS to perform the statistical analysis.[1] If you need to collect data in the field, you may find it convenient to use a data collection package that will run on a laptop computer, and later convert the data for analysis in SAS.

How Are Agency Data Stored?

In many assignments, existing data sources are needed to satisfy assignment objectives. Converting existing data from one format to another involves additional programming and risks that are best avoided. If your assignment needs existing data that are stored in a SAS format, you should copy the SAS data file and use it without converting it to another format. Conversely, if the data are stored as an SPSS data file, SPSS is a better choice than SAS, all other factors being equal.[2]

Is the Staff Proficient in Using SAS?

To meet the staff qualifications requirement of the Government Auditing Standards, the assignment team must collectively have the skills to use SAS. The supervisor or a designated person has to be able to review the SAS programming and understand the principles behind the SAS procedures. In addition, the

[1] dBASE is the registered trademark of Borland International, Inc., for data base software.

[2] SPSS is a registered trademark of SPSS, Inc., for its computer software product.

referencer must be able to reference workpapers produced with SAS. If the referencer does not possess the necessary skills, a person independent of the assignment and familiar with SAS must review the work.

Does the Assignment Team Have Access to Computer Resources Capable of Processing SAS Programs?

You need SAS software to interpret your SAS programs and to process your data. Since SAS is sold as modules, you must have the appropriate modules. For example, if your assignment calls for economic time series analysis, you will need the SAS econometric and time series module.

The SAS software must reside on a computer with enough capacity to process your data. If a large agency data file is to be processed, you need access to a mainframe. If a small questionnaire is to be processed, access to a microcomputer with SAS is sufficient. Depending on your processing needs and resources, you may use a combination.

Not all of GAO's microcomputers have enough speed and storage capacity to run SAS effectively. As a minimum, you need a computer with 20 megabytes of free hard disk space. The SAS Institute recommends a "286" or better computer.

Besides an appropriately sized computer for SAS software, other factors should be considered. For example, if you need to process classified data, the computer must be certified for processing classified data. If you need computer graphics, the computer must have access to a graphics printer.

Does the Assignment Team Have Access to SAS Support Material?

SAS reference manuals must be available. These materials are needed during the assignment to properly implement and understand SAS procedures. While the SAS "help" and the SAS "assist" facilities provide limited assistance, you need the SAS reference manuals for a complete understanding. If manuals are not available, they should be purchased.

SAS Work Plan

As encouraged by the Project Manual, your assignment team needs to decide on an initial SAS work plan. This plan should be a part of the overall assignment plan or evaluation design. The SAS work plan is important since it outlines the methodologies, programs, and procedures that are needed and shows what data will be captured and verified. It logically ties together the information needed, programs, data files, and reports.

The SAS work plan may include a process flow chart or diagram supported by a written narrative explaining the relationship among major programs, procedures, windows, data files, manual procedures, and reports. Consider the following items for your plan:

- how data relevance and reliability will be gathered, transferred, or converted to a machine-readable form;
- how data will be checked;
- when, if ever, a permanent SAS data file will be created;
- when and how data will be transferred between SAS and other programming packages;
- what major analysis procedures will be used and how they relate to the assignment objectives;
- how programs will be tested and reviewed;
- what major reports and data files will be produced; and
- who will be responsible for the analysis and review.

Upon arriving at your SAS work plan, you may have rejected many alternative approaches. You should document the rejected approaches in the plan. This will help you to justify your decisions and, if conditions change, help you to revise the plan.

Chapter 3

Checking Your Data and SAS Programs

Quality evidence and sound analysis are necessary to support findings, conclusions, and recommendations. To ensure the quality of your evidence and the soundness of your analysis in SAS-based work, you must use data-checking techniques and program verification procedures that are tailored to SAS. This section discusses a variety of such techniques and procedures.

Checking Your Data

The Government Auditing Standards suggests checking the relevance and reliability of data. Data reliability can be determined by conducting a review of the general and application controls in computer-based systems or by conducting other tests and procedures. SAS programs are an excellent tool for checking data since SAS contains procedures that can easily highlight potentially invalid data.

The nature of your data checks depends largely on how you answer the following questions:

- What is the source of your data file? That is, did the agency supply the data file, did you develop it, or was it program generated?
- How is the data file structured? Does the file have embedded structure information relating records? Does the file consist of records of a single record type?
- How are the individual data elements stored? Are they stored as numbers or as alphabetic characters?
- How many data values are associated with each data element? In other words, is the data element continuous, or is it categorical with a limited number of values?
- Can you check your data against other available data?

The extent of checking will depend on the risk associated with using the data. The following suggestions will help you use SAS to check data.

Reasonableness Checks

- Extreme values in numeric data variables may indicate invalid data. For example, the indicated ages of the youngest and oldest employees may be errors. You can identify extreme values with the PROC UNIVARIATE procedure. This procedure will report the five minimum and five maximum values of any data variable and will identify the observations from which the data came. Code the request in the following manner:

```
          PROC UNIVARIATE;
               VAR NUMVAR;
               ID GAOID;


     where NUMVAR   is the numeric variable being checked
                    and
           GAOID    is the identifier for the observation.
```

- Many numeric variables have only a limited range of valid values. For example, the age of an employee is normally greater than 15 years and less than 80. You can identify data outside an acceptable range with the IF ... THEN PUT ... statements. Information on the potentially invalid values will appear on the SAS log file. Code the request in the following manner:

```
      IF (NUMVAR LT MINVAL) OR (NUMVAR GT MAXVAL) THEN
           PUT NUMVAR= GAOID= 'Out of range';

where NUMVAR    is the numeric variable being
                checked,
      MINVAL    is the minimum acceptable value,
      MAXVAL    is the maximum acceptable value, and
      GAOID     is the identifier for the
                observation.
```

- Categorical data normally have few valid values. For example, the gender of a person coded other than M or F is a potential error. You can identify all values of a categorical variable in a data file using the PROC FREQ procedure. PROC FREQ counts the number of observations for each and every value of the variable. Code the request in the following manner:

```
        PROC FREQ;
             TABLES CATVAR;

where CATVAR    is the categorical variable being
                checked.
```

- You can identify invalid categorical data and the observations from which the data came with the IF ... THEN PUT ... statement and the IN function. Code the request in the following manner:

```
          IF CATVAR NOT IN('VAL1','VAL2','VAL3') THEN
               PUT CATVAR= GAOID= 'Unacceptable value';

where CATVAR   is the categorical variable being
               checked,
      VAL1, VAL2, and VAL3
               are the acceptable values, and
      GAOID    is the identifier for the observation.
```

- SAS will ensure that numeric data are numeric and that dates and times are valid. SAS does this while reading raw data. If bad numeric data or invalid dates and times are found, SAS will report the data on the SAS log and will assign a missing value to the variable in the observation. For each observation with an error, SAS will display the entire input record, the values assigned to each variable, and error messages. Do not suppress error-checking messages. This is important in checking the reasonableness of your data.
- SAS will help you check the consistency between data elements. Elements with conflicting values can be checked with IF ... THEN PUT ... statements. Code the request in the following manner:

```
          IF (NUMVAR1 EQ 1 AND NUMVAR2 EQ 5) THEN
               PUT NUMVAR1= NUMVAR2= GAOID=
               'Answers inconsistent';

where NUMVAR1  is a variable,
      NUMVAR2  is another variable, and
      GAOID    is the identifier for the observation.
```

- SAS will help you select a sample of data for you to compare for consistency with your source file. For example, you may want to select 10 observations from the beginning of the file, 10 observations from the end of the file, and 5 percent randomly from a SAS file of 10,010 observations. You can use an IF statement as follows:

```
               IF (_N_ LT 10) OR
                  (_N_ GT 10000) OR
                  (RANUNI(0) LT .05);
               PROC PRINT;

     where _N_ is a SAS variable to count
               observations and
           RANUNI is a SAS random number generator.
```

Missing Data Checks

SAS has two general types of data—numeric and alphanumeric. For numeric data, which include dates and times, SAS represents missing values with a period. For alphanumeric data, SAS represents missing characters with a blank.

- SAS will check for missing data in numeric variables with the PROC MEANS or PROC UNIVARIATE procedure. These procedures will report the number of observations with missing values. Code the request in the following manner:

```
          PROC UNIVARIATE;
               VAR NUMVAR;

     where NUMVAR   is the numeric variable being
                    checked.
```

- You can identify missing numeric data and the observations in which they appear using the IF ... THEN PUT ... statement. Code the request in the following manner:

```
          IF NUMVAR EQ . THEN
               PUT GAOID= 'Missing value';

     where NUMVAR   is the numeric variable being checked,
           .        indicates missing, and
           GAOID    is the identifier for the observation.
```

- SAS will check for missing data in categorical variables every time you use the PROC FREQ procedure. In the following example, all values, including missing values, will be identified. Code the request in the following manner:

```
          PROC FREQ;
               TABLES CATVAR /MISSING;

     where CATVAR   is the categorical variable being
                    checked.
```

- You can identify missing alphanumeric data and the observations in which they appear using the IF ... THEN PUT ... statement. Code the request in the following manner:

```
          IF ALPHANUM EQ '' THEN
               PUT GAOID= 'Missing value';

     where ALPHANUM is the categorical variable being
                    checked,
           ''       indicates missing, and
           GAOID    is the identifier for the observation.
```

Record Error Checks

- SAS will help you guard against losing observations by counting the number of records read and the number of observations created in a newly created data file. This information will appear on the SAS log file. You can check the number on the SAS log file against the expected number.
- SAS can help you locate unexpected duplicate observations. You can locate duplicates by first sorting the file with a PROC SORT procedure and then using an IF ... THEN PUT ... statement with the LAST and FIRST modifiers on the variable. Code the request in the following manner:

```
      PROC SORT;
           BY GAOID;

      DATA; SET;
           BY GAOID;
           IF NOT (LAST.GAOID AND FIRST.GAOID) THEN
                PUT GAOID= 'Possible duplicate';

where GAOID     is the variable where a duplicate value
                is possible,
  FIRST.GAOID   is a variable indicating the GAOID is
                the first observation in a group with
                the same value of GAOID, and
  LAST.GAOID    is a variable indicating the GAOID is
                the last observation with the same value
                of GAOID.
```

If duplicate observations exist, they will not be both the first and last observation with a particular value. For the above example, every duplicate will be listed on the SAS log.

The PROC SORT procedure using the NODUPKEY modifier eliminates observations with the same sort key. However, this method does not indicate which observations were deleted and should not normally be used in GAO work.

Another way to locate duplicate observations is to use the LAG function. First sort the file with a PROC SORT procedure and then using an IF ... THEN ... PUT statement and the LAG function code the request in the following manner:

```
     PROC SORT;
          BY GAOID;

     DATA;       SET;
          IF GAOID EQ LAG(GAOID) THEN
               PUT GAOID= 'Possible duplicate';

where GAOID    is the variable where a duplicate value
               is possible,
LAG(GAOID)     is the value of GAOID in the previous
               record.
```

- When an individual observation is structured as multiple records or lines, you can resynchronize the input after a missing record or a missing line by using the LOSTCARD statement. The LOSTCARD statement reports unexpected changes in the record identifier and is used in conjunction with the IF ... THEN PUT ... statements. The input file must be ordered by the record identifier. In the following example, the input data file has two records per observation:

```
          DATA;
               INFILE RAWDATA;
               INPUT GAOID1 NUMVAR1 #2 GAOID2 NUMVAR2;
               IF GAOID1 NE GAOID2 THEN
                  DO;
                     PUT GAOID1= GAOID2= 'Record error';
                     LOSTCARD;
                  END;

     where NUMVAR1  is a variable,
           NUMVAR2  is another variable,
           GAOID1   is the identifier for the observation on
                    the first record, and
           GAOID2   is the identifier for the observation on
                    the second record.
```

Checking Your Program

A program is reliable if it is performing as expected. SAS programs depend on several simple practices to enhance program reliability and error detection and to reduce the time to correct errors.

Before writing a SAS program, you should establish its specific purpose. Include a general description of your program and the techniques you will use to verify that the program is working correctly. These details will assist in selecting the appropriate SAS statements and procedures, as discussed below.

Program Structure

- Dedicate a program to one main purpose. This may seem inefficient, which it is for production programs where programs are run repetitively. However, by limiting your program to one purpose, it is easier to test, correct, and understand. If you have a complex analysis that takes several steps to achieve one purpose, breaking up the analysis into more than one program may make more sense.

- Preprogrammed procedures and built-in formats are more likely to be error-free than programs and formats you write. If a preprogrammed procedure or built-in function exists and fits your analysis needs, use it. For example, the descriptive statistics produced by the PROC MEANS procedure can be duplicated using a DATA step; however, the PROC MEANS procedure takes less time to program and gives reliable results.
- SAS statements and procedures are reliable, but they have limitations and known errors. Before writing SAS programs with unfamiliar statements or before using unfamiliar procedures, read the appropriate sections of the SAS manuals and check the usage notes for any limitations.
- Keep SAS data steps simple and straightforward by using standard programming constructs which are easy to follow. Avoid complex and unstructured programming which is characterized by excessive GOTO statements, too many levels of nested DO loops, self-modifying code, excessive interaction between modules, and multiple modules performing the same function.
- You should let programs run to their logical end. Use modular programming constructions like the DO WHILE and DO UNTIL statements. Each construction should have only one entry and only one exit point.

Readability

- Individual data steps and procedure steps are logically distinct parts in a SAS program and should be visibly separated in a program by at least one blank line and a description at the beginning of a new process. A RUN statement at the end of each step will visibly separate SAS programming steps and will assist in correcting errors.
- Put a program description near the beginning of every SAS program. Put a description of the SAS data step

or procedure step before every major logical grouping. At the beginning of the program, you can create a comment block with the following information: job title and assignment number, programmer's name, program name and date prepared, program description or purpose, and data source. For example:

```
***************************************
*       Review of Medical Costs       *
*               <123456>              *
*                                     *
*    Programmer:  J.D. Programmer     *
*                                     *
*    Program :  MEDCST01.SAS          *
*               August 31, 1992       *
*                                     *
*    This program edits the cost file *
*    for bad entry codes.  Bad codes  *
*    are listed on a print file for   *
*    manual validation.               *
*                                     *
*    Source:  MEDCT000 Cost File      *
***************************************;
```

At the beginning of each major logical grouping in the SAS program, create a comment block containing the purpose of the following step. For example:

```
***************************************
*                                     *
*     This procedure prints the       *
*     sample of individuals in SSNs   *
*     for account verification.       *
*                                     *
***************************************;
```

Do not explain the meaning of SAS procedures or statements in the program since they are explained in SAS manuals.

- Write meaningful comments into the source code. This will help the reviewer understand your logic.

  Comments should be easily distinguished from the program code. To write comments directly into SAS programs, use the * statement or the /* */ comment delimiter. For larger comments, draw boxes of asterisks around them. Avoid burying the /* */ comment delimiters within a SAS statement since the delimiters can make the program harder to read. Avoid using the /* in the first columns of a program because certain IBM operating systems will confuse the delimiter with the job termination statement.[1]

[1] IBM is the registered trademark of International Business Machines Corporation.

- Put global titles near the beginning of the program. The first title line, TITLE1, will contain the assignment name; the second line will contain the assignment code. These titles will remain constant throughout the SAS program. Every procedure that produces output will have an associated title which explains the output. For example:

```
TITLE1 'Review of Medical Costs';
TITLE2 '<123456>';

PROC PRINT;
   WHERE SSN EQ '' AND FY EQ 91;
TITLE4 'Records Lacking Social Security Numbers';
TITLE5 'for Fiscal Year 1991';
```

- When possible, enhance SAS output with descriptive variable labels and descriptive data formats. The LABEL statement associates a variable name with a label. The FORMAT statement associates data values with a description.
- Correcting errors and understanding data step programs is easier if logical groupings are readily apparent. Likewise, correcting errors and understanding procedures is easier if the procedure modifiers and options are logically arranged. The example below illustrates the lack of logical groupings in a data step:

```
DATA; SET;
 IF VAR4 EQ 3; IF VAR1 LE 6; IF VAR2 EQ 4;
 IF VAR1 GE 3;
 IF VAR1 GT .;
```

The example above is rewritten below to show the advantage of logically grouping statements in a data step.

```
DATA; SET;
 IF VAR1 GE 3 AND VAR1 LE 6;
 IF VAR2 EQ 4;
 IF VAR4 EQ 3;
```

- Indentation will also help in understanding and correcting programs. For example, indent all statements in a DO group at least one column from the word DO. Put the corresponding END statement on a separate line and indent it at least one column from the word DO, as follows:

```
DATA; SET;
 DO WHILE (VAR1 EQ 3);
  PUT VAR2=;
  END;
```

As another example, indent all statements in a SELECT group at least one column from the word SELECT. Put the corresponding END statement on a separate line indented one column from the word SELECT. Indent the OTHERWISE statement one column from the SELECT statement.

```
DATA; SET;
 SELECT (VAR1);
  WHEN (3) PUT VAR2=;
  OTHERWISE;
  END;
```

Indent all statements in a MACRO routine at least one column from the MACRO header line. Put the corresponding MEND statement on a separate line indented one column from the label.

```
%MACRO PNT;
  PROC PRINT;
 %MEND PNT;
```

- Align items for readability. The following two program examples look exactly the same to SAS, but the latter is more readable.

```
DATA; INFILE RAWDATA;
INPUT @17 VAR1 5. #2  @3  VAR2 MMDDYY8. @15 VAR3
5.2; IF VAR1 LT VAR3 AND VAR1 NE . THEN VAR1=VAR3; RUN;
PROC PRINT; RUN;




DATA;
     INFILE RAWDATA;
     INPUT @17 VAR1 5.
       #2  @3  VAR2 MMDDYY8.
           @15 VAR3 5.2;
  IF VAR1 LT VAR3 AND VAR1 NE .
     THEN VAR1=VAR3;
RUN;
PROC PRINT;
RUN;
```

- The IF statement may be written on one line. When more than one line is required, put the word THEN and the action clause on the line or lines following the condition clause. Indent the line or lines at least one column from the IF statement. Indent the ELSE statement at least one column from the IF statement on a separate line.
- The SAS MENU facility creates a special problem since its spacing is very different from what this guide suggests. When using code written by the SAS MENU facility, realign the code to comply with standard indentation and spacing.
- Use descriptive and consistent program names. If possible, relate the first part of the program name to the assignment or assignment segment. Use the extension "SAS" for the last level of the program name to indicate that the program is a SAS program. For example, MEDCST01.SAS is program number one for the medical cost review.
- File references are more understandable if they have meaningful names. Personal names of assignment team members are not acceptable. It is sometimes convenient to name the SAS file reference the same as the external file.
- Library references are more understandable if they have meaningful names. The library reference may be the same as the external file name.
- Programs are more understandable if variables have meaningful names or are related to other workpapers. For questionnaires and data collection instruments, you should tie variable names to items on the questionnaires. For example, a variable named Q12b would correspond to question 12b. For data coming from agency files, you can relate a variable to an agency data base. For example, a variable named ENTRY123 would correspond to a variable in an agency data dictionary.

In certain situations, variable names can be related to the nature of an item. For example, a variable named AGE can be used for a person's age. Since SAS variable names are limited to eight characters, this method of naming variables is not appropriate when the variable cannot be described adequately within eight characters.

- Whenever possible, spell out comparison operators rather than using special symbols. For example, use LT rather than the < symbol. Printers are inconsistent in the way they print special symbols, and mnemonics are easier to understand than symbols. Also, symbols do not have the same meaning in all programming languages. For example, < > means the maximum of two values in SAS, while it means not equal in BASIC.
- Operator priority in a complex expression can be confusing. When this is the case, use parentheses to show the order in which operations are performed.
- Use decimal numbers whenever appropriate, since in most situations they are more widely used and understood than binary and hexadecimal numbers.
- Avoid nonstandard coding schemes. The Federal Information Processing Standards Publications are a good source of standard codes on which many SAS functions and formats are based. For example, all SAS state code functions and United States maps are based on the Federal Information Processing Standards for state abbreviations.

Logic and Statistical Correctness

- Explicitly document options you use when they differ from the standard normal default settings. Use the OPTIONS statement to set options. Unexplained option selections may change the way a SAS program operates.
- When defining categories or groupings for SAS procedures or programs, make sure you have explicitly defined the categories as mutually exclusive

and collectively exhaustive. When using the IF and SELECT statements, make sure all possibilities are stated and the entire range of values is covered. Be careful that the end points of the categories reflect assignment needs and do not overlap.

Be explicit about all conditions, even those that may be unusual. Use the ELSE statement with the IF statement. Use the OTHERWISE statement with the SELECT statement. Use the special range names–HIGH, LOW, OTHER–in the FORMAT procedure. In the following examples you are distinguishing between gender and allowing for unusual responses.

```
DATA FEMALE MALE ERROR;  SET;
 IF GENDER EQ 'F' THEN OUTPUT FEMALE;
  ELSE IF GENDER EQ 'M' THEN OUTPUT MALE;
  ELSE OUTPUT ERROR;

DATA FEMALE MALE ERROR;  SET;
 SELECT (GENDER);
  WHEN ('F') OUTPUT  FEMALE;
  WHEN ('M') OUTPUT  MALE;
  OTHERWISE  OUTPUT  ERROR;
 END;
```

- Statistical procedures may be defined for either samples or universes. When sample statistics are desired, use the correct options. For example, in PROC MEANS, the option VARDEF = DF requests that the procedure use the proper degrees of freedom for computing the sample variance.
- Frequently, one SAS observation represents multiple observations. To deal with these situations, SAS has two statement modifiers–FREQ and WEIGHT. Do not

confuse them. FREQ is used for integer weighing. WEIGHT is used for any value greater than zero.

- SAS procedures do not calculate a weighted skewness or kurtosis. The WEIGHT statement affects only the mean, variance, and related statistics.
- SAS procedures use standard algorithms for computing sample statistics. SAS also provides options to test for normal distributions of the data. You must decide whether the standard algorithms are appropriate. The UNIVARIATE procedure's NORMAL option will test if your data values approximate a normal distribution, but you must decide which statistics are valid.
- Program explicitly for missing and miscoded data. In the FORMAT procedures, use the OTHER modifier to allow for missing and miscoded data. In procedure steps, select the MISS or NOMISS modifiers as needed. Use the SUM statement when missing data are to be excluded from the calculation.

Program Testing

- To ensure that your program is logically correct, review each program step to ensure there are no logic errors. If the program is large or quite complex, ask your supervisor to review the program with you in detail. If your supervisor needs assistance, a knowledgeable colleague can help.
- Test SAS procedures. One method is with small test files, comparing the SAS results with known results. Include bad, missing, and valid data in your test files. Another method is independently testing results manually or through another package, again comparing the two results. Test results should be documented and included in the workpapers, as well as an explanation of your choice in the use of options and modifiers.
- Once a program receives approval, do not modify it unless the modification is approved.

Efficiency

- Although effectiveness rather than efficiency is the primary concern in our programming environment, avoid excessively inefficient programming. Minimize computer time by not repeatedly reading data. For example:

```
PROC PLOT;
     PLOT A*B;
     PLOT C*D;
```

is preferable to

```
PROC PLOT;
     PLOT A*B;
RUN;
PROC PLOT;
     PLOT C*D;
```

In the second case, SAS must read the file twice.

- Avoid repeating complex and time-consuming calculations and file sorting. After making a calculation that you intend to use in a later program, save the calculation as a variable in the SAS data file. Avoid repeatedly sorting a file by saving the file in the order that will be most useful in future programs. Indexing may also be efficient.
- Conserve computer storage by eliminating variables with no further use.
- SAS programs may be more efficient if they temporarily limit the number of variables being processed. This is accomplished with the DROP or KEEP options attached to the SET statement. For example:

```
DATA; SET OLD(KEEP=VAR1);
  IF VAR1 GE 3 AND VAR1 LE 6;
```

- For numeric variables requiring less than 16 digits of precision, you can reduce the space SAS normally uses to store data by using the LENGTH statement.

Chapter 4

Data Entry and Transfer

The method you use to read data into a SAS data set will depend on the data source. In every case, you must verify that the data were entered correctly. These requirements are discussed in the Government Auditing Standards. This section explains how to implement those requirements with SAS.

Raw Data

In the data step, the INPUT statement is used for reading raw computer data. Most data from agencies, questionnaires, and data collection instruments are converted to SAS with the INPUT statement. SAS can read any file and record structure, no matter how complex. The following points will help you input the data:

- To reliably enter data, use a complete description of the data file, including a file structure, a record layout, and data descriptions for at least the critical data elements.
- The reliability of the data entry program is critical. Compare the resulting values of the SAS data set with raw data to ensure that data conversion is reliable. If the data entry program is at all complex, test it with a test file and document the results.

  Sources of errors may be (1) incorrectly specifying a data field starting location and length, (2) misinterpreting the format of the raw data, and (3) exceeding the field length of a SAS variable.

- Document any differences between the raw data file and the newly created SAS file. If records are deleted, document the number of deleted records and explain why they were deleted. The raw data record count must equal the sum of the SAS observations created and the records deleted. Data base segments must be counted when the raw data file is a data base.
- Fixed-format data entry should be used in preference to free-formatted and named data entry. Free-formatted data entry can lead to misread and

skipped data. When free-formatted data entry is the only alternative, you should count the number of data elements read for each observation.

- Maintain consistency between the variable and its format. To do this, completely define the variable attributes. Use either the ATTRIB statement with the FORMAT, INFORMAT, LABEL, and LENGTH specifications or individual statements for FORMAT and LABEL with the input format defined in the INPUT statement. When using the INPUT statement in the data step, use explicit input formats and read controls. Avoid suppressing input error reporting.
- The LENGTH statement is especially important for character variables since letting SAS decide their length could cause truncation. The maximum length of a SAS character variable is 200 characters. If you need a longer character variable, you have to either split the variable into parts or use another programming package.

  The LENGTH statement is also used to specify the number of bytes set aside to store the numeric variables. The maximum length and the default length of numeric variables in SAS is eight bytes, which equates to the integer 75,057,594,037,927,935. If you need more precision, you will have to split the variable into parts or use another programming package.

- Be careful with short records. When necessary, use the MISSOVER modifier on the INFILE statement. Use the LOSTCARD statement to synchronize data on multicard input.

SAS Data

Some agencies maintain data in a SAS file format. SAS can read these files without error. However, be aware of the following points:

- You must have a complete description of the data file and data descriptions for the critical data elements.

You must also get a copy of agency-written SAS formats. Use the PROC CONTENTS procedure for a description of the file and how it was created.

- You must know the SAS version used to create the data file and how it was exported.

Data Formats From Other Statistical Software Packages

The file you need may be in the native format of another statistical software package, such as SPSS or BMDP.[1] While SAS can read these files, you should keep the following points in mind:

- Be careful that the conversion maintains consistency between the value of the variable and its format. Be cautious of variable name conversions, truncation of long character variables, redefinition of missing data, changes in date conventions, and changes due to format inconsistencies. These potential inconsistencies are documented in the language and procedure manuals.
- Have a complete description of the agency's data file, including a file description and data descriptions of all critical data elements.
- Test and document the reliability of the data conversion process. Trace the data from the other statistical package's data file to the SAS data. Sources of error may be mistakes in converting the variable name, field width changes, and data format differences. Date formats are especially troublesome.
- The data entry program must read every observation in the other package's data file and create a SAS image. If records are deleted, the number of deleted records should be reported. The input observation record count should equal the sum of the SAS observations created and the records deleted. Data base segments should be counted when the raw data file is a data base.

[1] BMDP is a registered trademark of BMDP Statistical Software, Inc.

Data From Data Base Packages

Some agencies maintain data using data base packages, such as dBASE. SAS can read these files with the PROC DBF procedure for dBASE files and with special procedures for other data bases. To convert data from data base packages, you should be aware of the following points:

- Have a complete description of the agency's data file and data descriptions for the critical data elements, as well as a description of the file structure.
- Test and document the reliability of the data conversion programs. Sources of error may be mistakes in record segment pointers and incorrect SAS record structures.

SAS-Generated Data

SAS can generate data internally in data and procedure steps. Data generated in this manner must have the same level of reliability as data from external sources. Several points deserve special mention.

- GAO has approved the use of the SAS RANUNI random number generation function and subroutine to generate uniform random numbers for sample selection. When you need more than one stream of random numbers, use the subroutine rather than the function.
- Be careful to use the correct random number function since SAS offers several functions with different output distributions. Avoid the UNIFORM random number generation function.
- Check character data created from character substrings for leading and trailing blanks.
- Numeric data created from mathematical calculations are subject to rounding errors. The ROUND and FUZZ functions will eliminate rounding errors at specified rounding units.

On-Line SAS Data Entry

You can manually enter data directly into SAS data sets using the CARDS option, direct variable assignment, and full screen products, such as PROC FSEDIT. Data entered in this manner must have the same level of reliability as data provided by other methods. Also, you must be careful to create a solid audit trail. Guidelines specific to this method of data entry follow:

- Avoid the CARDS and CARDS4 methods of data entry since they require the data and the SAS program to be ready, reviewed, and approved at the same time. These methods are difficult to use because the programmer is not normally the same person who enters the data. In addition, the SAS log file does not show the data lines in the program, so the program must be included in the workpapers to document the raw data.
- The SAS assignment statement is not a good way to enter a lot of data. It is normally used in a batch mode for correcting data in an existing SAS data file. A program using the assignment statement must carefully document the data.
- Data can be entered directly into SAS files with the PROC FSEDIT procedure or by creating data entry windows. These methods have the advantage of immediate data editing, but a danger exists in that the data file can be lost or inadvertently modified. Always make a backup copy of the data file before using this method.
- The PROC FSEDIT procedure does not have a log file of the changes made to the data base. Use the PROC COMPARE procedure to compare values between the old and new data files, and review the output report.

Transferring and Moving SAS Data Sets

Transferring and moving SAS data sets is reliable, but specific methods must be used to retain the integrity of the files. Several methods are available, as discussed below:

- When copying SAS data files within a computer system, use the PROC COPY procedure. Do not use system utilities, such as the mainframe computer's IEBCOPY program or the microcomputer's COPY procedure. These utilities will not maintain the integrity of the SAS file.
- You can use the interactive CATALOG window to copy catalog entries from one catalog to another. Catalog entries include windows, formats, indices, and many other SAS items. You can use the interactive DATASETS procedure to copy SAS data files.
- When copying files between remotely linked computers running SAS, use the PROC UPLOAD procedure or the PROC DOWNLOAD procedure.
- When copying SAS files between computer systems, use the PROC COPY procedure with the IMPORT or EXPORT options.

Chapter 5

Processing and Documenting the Results

Documentation standards for SAS programs are similar to those for manual workpapers; that is, SAS workpapers should be complete, accurate, clear, neat, relevant, and understandable. The term "understandable" needs some clarification when dealing with SAS programs. SAS programs must be understandable to people with a general knowledge of SAS. However, SAS documentation need not be a complete explanation of SAS procedures and statements since this information is available in the SAS language manuals.

Documentation of a SAS program helps ensure the quality of the product and provides the supervisor and the referencer with an essential audit trail. The proper time to document your SAS program is during program development since documentation also helps you ensure that your program is working correctly.

Documentation includes the SAS log file and the output. The documentation can also include, but is not limited to, flow charts, test results, comments and titles within the programs, and explanations of messages in the SAS log file.

All GAO workpapers should receive prompt supervisory review, and workpapers produced by SAS are no exception. The supervisor should indicate review and acceptance of the SAS program by signing on the first page of the program output. For output from mainframe computers, the first page is the job control output log. For microcomputers, the first page is the SAS log file.

The following suggestions help ensure that SAS programs will be developed with sufficient documentation and adequate review.

- You are encouraged to use interactive programming for program development, but workpapers for the final programs should be run in the batch mode or with a clean interactive run.

- SAS programs are easily modified. To accurately document the running of a SAS program you must retain the log file in the workpapers. SAS output reports must never be separated from their log file. Formal run books are not needed. On mainframe versions of SAS, the job control language messages should also be retained.
- The log file must be complete, including all source file statements. When calling external programs or macros with the %INCLUDE statement, use the SOURCE2 option so that all source file statements are listed on the SAS log file. When using macros, use the MPRINT and SYBOLGEN options to show the macro and to display how symbolic variables are resolved.
- Messages on the SAS log should be reviewed and, where appropriate, explained by the programmer. Log file notes needing comment are data read errors, format conversion errors, format changes, missing value generation, record counts, and variable counts.
- Output reports must be complete. Use the default options for date and page numbers.
- Systems documentation must indicate final program names, data file locations, and workpaper indexes.
- SAS files and user-written permanent formats can be documented with the PROC CONTENTS procedure. Use the PROC PRINT procedure to display a portion of permanent SAS data files.
- Run final test programs in the batch mode or as separate interactive runs. The resulting workpapers require approval of a knowledgeable person, preferably the supervisor.
- If possible, limit procedure output to relevant information. For example, if cell percentages are not needed in frequency tables, suppress printing the percentages with the NOPERCENT option. Where extraneous material is part of the output, line it out.

- Avoid using the TITLES window since this method of entering titles does not create an audit trail. Likewise, avoid using the FOOTNOTES window for entering footnotes for the same reason.

Referencing SAS Work

Full referencing is the preferred method for all GAO reports, testimonies, and other products, and material based on SAS is no exception.

To reference SAS-based information, the referencer must be able to determine that the workpapers provide sufficient, competent, and relevant evidence. Referencers who are not familiar with SAS must seek assistance from someone who is. This person must also be independent of the assignment.

The following guidance will help the referencer reference SAS workpapers:

- The referencer is responsible for determining that the SAS workpapers are in compliance with GAO workpaper standards. For example, the SAS workpapers need evidence of supervisory review and quality control checks. The position of the titles, preparer's name, date, supervisor's signature, and source reference may vary, but they are required.
- The referencer must check the SAS runs for unresolved errors, unexplained changes in the number of variables, unexplained changes in the number of records, unexplained variable type conversion, and unexplained generation of missing data. The SAS log file will contain many of these items.
- The referencer must determine if the workpapers include cross-referenced system documentation; program documentation, including log files; data documentation and validity tests; and tests of complex programs.
- The workpapers need sufficient instructions so that the referencer can duplicate the work. Instead of manually verifying every computation (which, in most cases, would be impossible), the referencer must ensure that computer program logic has been tested, the data have been checked for reliability, and the report item matches the supporting SAS output. On

rare occasions, the figures may be independently verified using another computer program.

- Factual data derived from SAS procedures must be accurate and objective. The procedures must have comments indicating the assumptions made and the method used. Items to check will depend on the procedures used. The following examples indicates typical items to check:
  - When referencing the PROC ANOVA procedure, check the use of an unbalanced experimental design and changes in the significance level from the standard 0.05. Deviations should have written justification.
  - When referencing the PROC CATMOD procedure, check the design matrix specification to ensure it is consistent with the results.
  - When referencing the PROC FREQ procedure, check the inclusion or exclusion of missing data in marginal computations, the proper adjustment to Chi-square tests, and warnings about minimal cell frequencies.
  - When referencing the PROC UNIVARIATE procedure, check the proper specification for the divisor in the calculating variance.
  - When graphs are based on SAS procedures, the referencer must ensure that the axes do not distort the data and that the smoothing techniques do not hide important changes.

Chapter 7

Disposition of Workpapers

Retention standards for SAS workpapers are the same as those for manual workpapers. The method of archiving data and programs will depend on the amount of data and the facilities available for archiving. Microcomputer data will most likely be archived on a floppy disk. Mainframe computer data will most likely be archived on magnetic tape.

You do not have to save every data file you create in the course of a job. The files that are archived will depend on how the data were created and what files would be time consuming to recreate. The following items should be considered in archiving SAS programs and data files:

- Use only SAS procedures, such as the PROC COPY procedure, and DATA steps to make backup copies of SAS data files. Never use operating system utilities on these files.
- When saving data files, remember to include formats, windows, and indices.
- When the SAS file is likely to be used on another system, backups should be saved as portable files using the EXPORT function.
- All final SAS program files must be saved. SAS programs should be saved as text files.

Additional Information

The SAS Institute publishes an extensive list of SAS software guides which are customized for various operating systems and computer environments. A few of the over 200 SAS publications are listed below.

Version 6 – MS-DOS and PC-DOS environment:

SAS Language Guide for Personal Computers, Release 6.03 Edition

SAS Procedures Guide, Release 6.03 Edition

SAS/STAT User's Guide, Release 6.03 Edition

SAS/GRAPH User's Guide, Release 6.03 Edition

SAS/FSP User's Guide, Release 6.03 Edition

Version 6 – AOS/VS, CMS, MVS, OS/2, PRIMOS, and VMS environments:

SAS Language: Reference Version 6

SAS/STAT User's Guide, Version 6

SAS/GRAPH Software Reference, Version 6, volumes 1 and 2

SAS/FSP Software Usage and Reference, Version 6

SAS Companion for the MVS Environment, Version 6

You will find additional guidance in these publications:

Federal Information Processing Standards Publications, Department of Commerce, National Bureau of Standards. Washington, D.C.: 1970-92.

Using Micro Computers in GAO Audits: Improving Quality and Productivity. General Accounting Office,

Information Management and Technology Division, Technical Guideline 1. Washington, D.C.: 1986.

Assessing the Reliability of Computer-Processed Data. General Accounting Office, Information Management and Technology Division, GAO/OP-8.1.3. Washington, D.C.: 1990.

Project Manual. General Accounting Office, chapter 10.1. Washington, D.C.: 1986.

Government Auditing Standards, 1988 Revision. General Accounting Office. Washington, D.C.: 1988.