AN INTERACTIVE FRONT-END FOR A DATA EDITING PROCEDURE USING SAS® WINDOWS

Lawrence R. Catlett, NIOSH
Lance W. Cameron, formerly with NIOSH

INTRODUCTION

We have previously described a method for validating questionnaire-based data using PROC FORMAT and a general purpose SAS macro.1 The method is descriptive rather than programmatic and is able to effectively handle complexities such as nested structures. However, the method has two serious disadvantages: 1) manually defining and coding the formats is time-consuming and tedious and 2) writing the code for the macro calls is likewise time-consuming and tedious. Our review of the method led us to believe that much of it could be automated. We also felt that if the method were to be used by researchers, a simple, interactive front-end would be required.

In this paper we describe an interactive, user-friendly front-end to automate the generation of formats and macro calls. We use SAS DISPLAY - WINDOWS and macro variables to create customized windows for data entry and use SAS to generate the SAS code for the necessary SAS FORMATS, as well as to produce the SAS code for the edit program.

This program was developed using an IBM/AT compatible personal computer running MS-DOS 3.3 and with 640K RAM. Version 6.3 of SAS/PC is required. Any machine which can run SAS/PC 6.3 should be able to accommodate the program. The resulting generated SAS code should be usable on a wide variety of hardware platforms.

BACKGROUND AND PROBLEM

The National Institute for Occupational Safety and Health (NIOSH) conducts research into the causes of occupational injuries, illnesses and diseases. Increasingly, data used for this research is collected through self-administered questionnaires by study participants. In many cases this work is done by contractors and we are simply given a computer tape with the resultant information. Consequently, NIOSH is interested in methods to verify data integrity. We developed a macro, %SKFCHK, which essentially eliminated the need for a programmer to write a unique edit FORMATS. These features allow the program to be written and tested very quickly.

METHOD

The DISPLAY, WINDOW and macro features are used to collect the information necessary to create unique edit FORMATS. These features also facilitate the termination and restart procedures in this program.

The process of defining the acceptable ranges for each question, although simple and straightforward, is repetitive, tedious and time-consuming. These are the factors that led us to believe that this application is an ideal candidate for automation. Automation of the data editing method must address several issues:

1) Collection of information that describes valid ranges and actions for each question;
2) Determination of a format for a given question;
3) Identification of unique formats;
4) Provision to allow the review/edit of previous input.
5) Compilation of all information required to build the macro call to actually edit the data.
6) Creation of the actual SAS code for the FORMATS and macro execution statements.

In addition, we wanted the program to be "universally" usable by all SAS users. For this reason, we chose to use base SAS with its DISPLAY, WINDOW and macro features to create a customized data entry screen. These features allowed the program to be written and tested very quickly.

DESCRIPTION OF THE DATA EDITING PROGRAM

This program is used in conjunction with the program %SKFCHK which was described above. The program uses the DISPLAY, WINDOW and macro features to create a customized data entry screen.

When the program is executed, a screen is displayed (refer to Figure 1), providing the user with some general information. In the next screen, Figure 2, the user is required to enter a "name" for the questionnaire. This name must be used in all future sessions. The program then creates a directory by this name.
on the A: drive; however, the drive designation can be changed in the program if desired. If you are unsure of the name you previously entered, you have the opportunity to look at the directory on the A: drive. All files relevant to this questionnaire are stored in this directory.

FIGURE 2

Enter the name for the questionnaire file

The next screen is a menu where the user decides whether to enter questionnaire data, edit previous entries or generate the programming code (refer to Figure 3). One nice feature of this program is that it allows the user to start and stop almost anywhere he/she chooses. For some of our longer questionnaires, it may take two or three days to enter this data.

FIGURE 3

The input mode window appears next (refer to Figure 5). This window has four types of input. The first, at the top of the window, is for the name assigned to the question. You also can see three input columns on this screen. The left column with the heading "LOWER-BOUND" is always filled.

After entering the last range, press the "PG DN" key to go to the next question. This program arbitrarily limits the user to eight values/ranges. Typically, four ranges has been sufficient for our questions. The number of range checks can be changed easily in the program.

Some questionnaires are too lengthy to be entered at one sitting. The user can terminate the session at anytime by entering "STOP" for the question name and pressing the "PG DN" key. The program saves the information entered during the current session by combining it with information previously entered. This data is stored on the A: drive in the directory created and named by the user (refer to Figure 2). Whenever the user enters

The middle column with the heading "UPPER-BOUND" is only filled if there is a continuous range of acceptable values and then it will contain the upper-bound of the range. The right column is only filled if the associated response indicates that a "skip to another question" is to occur. In this case, it will contain the name of the question to skip to; otherwise, the field is left blank.

These entries are edited by the program. First, the question name is checked to ensure that it is a valid SAS variable name. Next, the lower and upper-bounds are checked for consistency. The values must be entered in ascending order to prevent the occurrence of a range conflict in the FORMAT being created (refer to Figure 5). When a question name or range conflict error occurs, a new screen appears telling the user an error has occurred and to press the "ENTER" key to enter the correct data and continue (refer to Figure 6).
"STOP" to terminate the input step, the program returns to the main menu where the user will be queried for the next step (refer to Figure 3).

To facilitate the ease of use of this program, we added an edit feature. This is Option 2 on the main menu. This option brings up another menu which allows you to print and review your previous entries or to modify the ranges of previous entries. The changes in this step are made to ranges and "skip to" variables only. The variable name cannot be changed.

The user should always carefully review the output logs from a session or job. Otherwise, the user cannot be sure that the session terminated with the desired results.

When you finally finish entering and editing the data, you are ready to generate the SAS code. Select Option 3 on the main menu to do this. A new window appears asking you to enter the variable name of the subject (refer to Figure 7). This name is a unique identifier on each questionnaire (input record) which allows you to go back to the source document to correct the error.

At this point, the program creates the SAS code for your edit program. First, it creates the FORMAT procedure. Note that only unique formats are created. Next, the code for the macro calls to the edit macro %SKIPCHK is created. The output is saved to the directory specified in Appendix 2. At this point, a knowledgeable SAS programmer can review this code and can enter the appropriate FILENAME and LIBNAME statements and include the %SKIPCHK macro. To illustrate this program, we have used the same sample questionnaire (refer to Appendix 1) and test data that appeared in our previous paper.

The program has its limitations and disadvantages. The edit programs created are limited by the capabilities of the edit macro %SKIPCHK and cannot handle special logic requirements. Question (variable) names cannot be changed. One fact remains, the process is still laborious.

Despite this fact, the program has several features to recommend its use. It is interactive and user friendly. The error checking on the input eliminates most errors. The program allows corrections to be made relatively easily and quickly. Perhaps the major advantage is that the process is straightforward and simple enough to allow someone with only basic data entry skills to do most of the work.

ACKNOWLEDGMENTS

Dr. Teresa Schnorr
Dr. Barbara Grajewski
Ms. Barbara Jenkins

REFERENCES


FOR FURTHER INFORMATION

Contact:
Lawrence R. Catlett
Chief, Support Services Branch
DSSRF-OSHSN
4676 Columbia Parkway
Cincinnati, OH 45226
APPENDIX I

SAMPLE QUESTIONNAIRE

Q1. Were you ever pregnant?
   1. Yes
   2. No (skip to Q6)
   3. Do not know
   4. Refused

Q2. Outcome of pregnancy #1?
   1. Live birth (skip to Q3a)
   2. Miscarriage (skip to Q4a)
   3. Other (skip to Q5a)

Q3a. Year of birth?
   1940-1988

Q3b. Born in a hospital?
   1. Yes
   2. No
   3. Do not know
   4. Refused (skip to Q6)

Q4a. Year of miscarriage?
   1940-1988

Q4b. Were you seeing a doctor at the time of the miscarriage?
   1. Yes
   2. No (skip to Q6)
   3. Do not know
   4. Refused (skip to Q6)

Q4c. May we contact your doctor for your medical records?
   1. Yes
   2. No
   3. Do not know
   4. Refused (skip to Q6)

Q5a. What was the outcome of the pregnancy?
   1. Induced abortion
   2. Stillborn
   3. Tubal pregnancy
   4. Currently pregnant

Q6. Are you currently employed?
   1. Yes
   2. No
   3. Do not know
   4. Refused

APPENDIX II

PROC FORMAT; * PROGRAM GENERATED FORMATS;

VALUE FRMT1_ = 1
2 = 2
3 = 3
. = 'M'
OTHER = 'E';

VALUE FRMT2_ = 'V'
2 = 'V'
8 -9 = 'V'
. = 'M'
OTHER = 'E';

VALUE FRMT3_ = 'V'
2 = 'V'
8 -9 = 'V'
. = 'M'
OTHER = 'E';

VALUE FRMT4_ = 'V'
2 = 'V'
8 -9 = 'V'
. = 'M'
OTHER = 'E';

VALUE FRMT5_ = 'V'
2 = 'V'
8 -9 = 'V'
. = 'M'
OTHER = 'E';

VALUE FRMT6_ = 'V'
2 = 'V'
8 -9 = 'V'
. = 'M'
OTHER = 'E';

VALUE FRMT7_ = 'V'
2 = 'V'
8 -9 = 'V'
. = 'M'
OTHER = 'E';

* REMEMBER TO COPY THE MACROS AND INCLUDE 
* FILENAME AND LIBNAME STATEMENTS AS NEEDED;

DATA NULL; * START OF SAS EDIT STATEMENTS;
SET ????;

* ???. FILL IN WITH DATA NAME;

RUN;