A GENERALIZED EDITING PROCEDURE FOR QUESTIONNAIRE-BASED DATA USING THE SAS SYSTEM

Lance W. Cameron, NIOSH
Lawrence R. Callett, NIOSH

I. INTRODUCTION

Developing programmatic procedures for validating questionnaire-based data can be very complex and time-consuming. We will describe a procedure using PROC FORMAT and a general purpose macro to accomplish this objective. In addition to generalizing the process, the procedure does not require high-level programming expertise.

II. BACKGROUND

The reliability of any scientific research is ultimately only as good as the accuracy of the data being analyzed. The National Institute for Occupational Safety and Health (NIOSH) conducts research in a number of areas focusing on causes of occupational injuries, illnesses and diseases. Much of the data used for this research is collected through the administration of questionnaires to study participants.

NIOSH recently initiated a study which involved collecting laboratory and questionnaire data. Two of these questionnaires were quite lengthy and complex. Although the data was collected by a contractor, NIOSH personnel had the responsibility of verifying the accuracy of this data, building the necessary data files and doing the data analysis.

Since SAS was designated for data analysis, we decided to use SAS for data editing and data management as well.

III. PROBLEM

Unlike editing laboratory data, which can be validated by blind verification and range checks, verifying questionnaire data poses an additional problem in that not all questions will be answered nor necessarily should be answered. A series of questions - for example, regarding smoking habits - are often asked only if the respondent's answer to a "gate" question indicates the relevancy of the questions - for instance, whether the respondent is a smoker. The respondent's answer to the gate question concerning smoking status will either direct the questioner to the series of smoking-related questions or will cause those questions to be skipped. This is referred to as skip patterns.

Because the sequence of responses follows different pathways depending upon the response to particular gate questions, editing questionnaire data tends to be more complicated. A program which edits questionnaire data must do three things. First, it must indicate when data is missing but should be present. Second, it must indicate when data is present but should be missing. And, third, it must indicate when the data is out of range.

One could establish special program logic to accommodate the skip pattern arrangements unique to each questionnaire, but this process would quickly get very complex and would require fairly sophisticated programming. A better alternative is to develop a general method for review of all questions such that the three above-mentioned editing requirements are accurately carried out. We develop one such method in this paper.

IV. Approach

The solution we developed for handling questionnaire data deemphasizes special program logic. Our goal is to simplify the programming effort and thus reduce development time and minimize the need for programming expertise.

The focus is upon defining the data using PROC FORMAT and processing all fields with a single defined macro. First, a format is defined for each general type of question; that is, for those questions with the same legitimate values and with the same actions associated with those values. The format lists all valid responses and assigns codes depending on what action is necessary. A 'V' codes indicates that the value is valid but does not invoke skipping; codes '1' through '8' indicate a skip which is to be terminated at the appropriate variable (discussed below); missing values are assigned 'M' and all other values are assigned 'E' for nonvalid responses. For example, suppose three
questions each allowed 1 (=NO) and 2 (=YES) as the only valid responses, and, further, a NO response required the interviewer to skip to other questions. All three of these questions would be filtered through the following format:

```
PROC FORMAT;
  VALUE AI I
  V
  M
  OTHER E
```

For each question (variable) under review, the macro %SKIPCHK is called and the following parameters are passed:

- **SUBJ**: the identifier for the observation (e.g., social security number);
- **QN**: the value of the variable under review;
- **FMT**: the format which applies to the variable under review;
- **S1-S8**: the values of the first to the eighth variables to terminate a "skip". In the above example, a NO response would skip until encountering the variable passed to the macro _S1 parameter. (The macro can easily be modified to allow more than eight skip variables).

In addition, the macro itself maintains the following variables:

- **SKIP**: the flag indicating whether skip is in effect;
- **FQN**: the name of the variable to end a skip;
- **QNAME**: the formatted value of the question under review;
- **ONAME**: then variable name of the question under review;
- **XI-XS**: the variable names for _S1 through _S8.

The variables QNAME and Xn are derived using a feature of the SAS macro language which allows the variable name to be extracted by placing quotes around a parameter.

For each question in sequence, the process works as follows:

**STEP 1**: The question under review is filtered through a format (defined in PROC FORMAT) to produce the variable FQN which will have the value 'V', 'M', 'E', or 'L'- 'B' (the codes discussed above).

**STEP 2**: If a skip is in effect then the macro %SKIPCHK compares the value of SKIP to the value of QNAME to see if the question under review is the question at which to terminate the skip. If the skip is not terminated then the data value must be missing else an error message is generated.

**STEP 3**: If the skip is terminated or no skip is in effect, then the formatted value of FQN must be checked as described above and must be 'V' or 'L'- 'B' else an error message is generated. If FQN is 'M' then the message indicates a missing value; if FQN is 'E' then the message indicates an invalid value. Codes 'L' through 'B' cause the SKIP flag to be set and SKIPQ to be set to the name of the variable to which one is to skip.

Several observations may be made with respect to the above description.

First, the approach is descriptive rather than programmatic. That is, the user need only be concerned with describing the question in the format definition and assigning the skip variables (_S1-_S8) in the macro call. The macro %SKIPCHK is sufficiently generalized to accommodate most cases without any special program logic.

Second, the macro itself satisfies all three requirements for editing questionnaire data: 1) skipped questions which should be missing but have values are logged; 2) questions which should have values but are missing are logged; and 3) out-of-range values are logged.

Finally, since all three general cases are handled by calling the macro for each question in sequence rather than by special program logic, complex structures such as "nested skip patterns" are accommodated in a straightforward manner.

One potential problem is that the number of formats needed could be quite large in a long questionnaire. However, we concluded from our experience with a several hundred variable questionnaire that far fewer formats were required than the number of questions because many questions were in the same format.

V. CONCLUSION

Questionnaire data can sometimes be quite convoluted and difficult to conceptualize. The approach we propose seems to substantially reduce the complexity of developing sound editing programs. Furthermore, we believe that the procedure can be used by individuals with limited SAS programming experience.
We have included an appendix with a simplified example questionnaire and an implementation of the technique proposed. Sample data is processed in an attempt to illustrate how the program handles different cases.

FOR FURTHER INFORMATION CONTACT:

Lance Cameron
Lawrence Catlett
National Institute for Occupational Safety and Health (NIOSH)
4676 Columbia Parkway MSR-4
Cincinnati, OH 45226

(513) 841-4244
APPENDIX

I. QUESTIONNAIRE

Q1. Were you ever pregnant?
   1. Yes
   2. No (skip to Q6)
   8. Don't know
   9. 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 hospital?
   1. Yes
   2. No
   8. Don't know
   9. 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)
   8. Don't know (skip to Q6)
   9. Refused (skip to Q6)

Q4c. May we contact your doctor for your medical records?
   1. Yes
   2. No
   8. Don't know
   9. Refused (skip to Q6)

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

Q6. Are you currently employed?
   1. Yes
   2. No
   8. Don't know
   9. Refused
II. SAMPLE PROGRAM

PROC FORMAT;
VALUE AA /* this format used for questions Q1 & Q4b */
1 = 'V' /* => 1 is valid and no skip to follow; */
2,8,9 = '1' /* => 2,8,9 are valid; skip to first "skip var"; */
. = 'M' /* => error...missing value; */
OTHER = 'E'; /* => all other values are erroneous! */

VALUE AB /* this format used for question Q2 */
1 = '1' /* => 1 is valid response & goto first "skip var" */
'2' /* => 2 is valid response & goto sec. "skip var" */
3 = '3' /* => 3 is valid response & goto 3rd "skip var" */
. = 'M' /* => error...missing value; */
OTHER = 'E'; /* => all other values are erroneous! */

VALUE AC /* no skipping indicated in the format AC; */
1940-1988 = 'V' /* => values between 1940 and 1988 are valid; */
. = 'M' /* => error because of missing value; */
OTHER = 'E'; /* => any other response is an error; */

VALUE AD /* 1,2,8,9 are valid & goto first "skip var" */
1 = '1' /* => 1,2,8,9 are valid & goto first "skip var" */
. = 'M' /* => error...missing value; */
OTHER = 'E'; /* => all other values are erroneous! */

VALUE AE
1-4 = 'V'
. = 'M'
OTHER = 'E';

VALUE AF
1,2,8,9 = 'V'
. = 'M'
OTHER = 'E';

DATA QUESTION;
INPUT ID Q1 Q2 Q3a Q3b Q4a Q4b Q4c Q5a Q6;
CARDS;
100 1 2 1950 1930 2 1
200 2 . 9 9
300 1 3 . 2
400 1 1 1998 . . . 7
500 1 3 . . 2 1
600 1 2 1979 1999 2 . . 1

%MACRO SKIPCHK(SUBJ, QN, FMT, __S1, __S2, __S3, __S4, __S5, __S6, __S7, __S8);
RETAIN OLD_ID 0 QNAME SKIP SKIPQ x!-.8 ' '; FILE 'A:ERR\MOD8.LST' HEADER=HDR PS=60; *output file for errors;
ARRAY _X_(B) $ x1-x8; * ARRAY USED TO SIMPLIFY CHECKING ARGUMENT LST;
******************************************************************************;
* SUBJ: ID;
* QN: ORIGINAL VALUE OF VARIABLE OF INTEREST;
* FMT: NAME OF THE FORMAT USED TO "VERIFY/EDIT" THE ABOVE VARIABLE;
* __S1-__S8: NAMES OF "SKIP TO" QUESTIONS FOR VARIABLE OF INTEREST;
* SPECIAL VARIABLES UNIQUE TO MACRO ";
* SKIP: FLAG INDICATING WHETHER SKIP PATTERN IN EFFECT;
* SKIPQ: NAME OF VARIABLE TO END SKIP - I.E., WHERE TO SKIP TO;
* FGN: FORMATTED VALUE OF VARIABLE OF INTEREST;
* QNAME: NAME OF VARIABLE OF INTEREST;
* X1-X8: VARIABLES CONTAINING THE NAMES(LITERAL STRING) OF __S1-__S8;
******************************************************************************;
IF &SUBJ NE OLD_ID THEN DO;
    OLD_ID = &SUBJ;
    SKIP="N";
    SKIPQ="N";
END;

x1="&_s1"; /* these statements "convert" variable names to labels/character strings */
x2="&_s2"; /* which are used below to determine where skip begins or ends */
x3="&_s3";
x4="&_s4";
x5="&_s5";
x6="&_s6";
x7="&_s7";
x8="&_s8";

DO I = 1 TO 8;
    IF SUBSTR(_X_(I),1,3) = '__S' THEN _X_'I);' /* THESE STATEMENTS ASSURE THAT ONLY A "PASSED" ARGUMENT IS USED. */
    END;
    /* NAME OF THE NEXT */

    "GIVES SKIPQ THE NAME OF THE "next* */
    SKIPQ = _X_(INPUT(FQN,2)); /* note that it is not neces- */
    END;
    this significantly reduces the amount of "typing" */

    "MISSING VALUE ";
    ELSE IF FQN = 'E' THEN PUT @3 &SUBJ @22 &QN @33 FQN @40 ' INVALID VALUE ';
ELSE DO;
    SKIP = 'Y';
    SKIPQ = _X_(INPUT(FQN,2)); /* GIVES SKIPQ THE NAME OF THE NEXT*/
    END;
    /* SKIP VARIABLE */

    "MISSING VALUE ';
    ELSE IF FQN EQ 'M' THEN PUT @3 &SUBJ @22 &QN @33 FQN @40 ' SHOULD BE MISSING ';
ELSE IF SKIP = 'N' THEN DO;
    IF FQN NE 'V' THEN DO;
        IF FQN EQ 'M' THEN PUT @3 &SUBJ @22 &QN @33 FQN @40 'MISSING VALUE ';
        ELSE IF FQN = 'E' THEN PUT @3 &SUBJ @22 &QN @33 FQN @40 'INVALID VALUE ';
    ELSE DO;
        SKIP = 'Y';
        SKIPQ = _X_(INPUT(FQN,2)); /* GIVES SKIP THE */
        END;
        /* NAME OF THE NEXT*/
    END;
    /* SKIP VARIABLE */

    "MISSING VALUE ';
    ELSE IF FQN NE 'M' THEN THEN PUT @3 &SUBJ @22 &QN @33 FQN @40 'SHOULD BE MISSING ';
    ELSE IF SKIP = QNAME THEN DO;
        SKIP = 'N';
        SKIPQ = 'N';
        END;

    "SAME OUTPUT OF EDIT ALGORITHM FOR SKIP PATTERNS' /339 'by' /325 'LANCE CAMERON AND LARRY CATLETT'///
311 'ERROR LOG' /3245 'SUBJECT' 312 'QUESTION' 322 'VALUE' 329 'FORMATTED' 345 'ERROR' 305 'ID' 314 'NAME' 331 'VALUE'/;
### III. Sample Output of Edit Algorithm for Skip Patterns

<table>
<thead>
<tr>
<th>SUBJECT ID</th>
<th>QUESTION NAME</th>
<th>VALUE</th>
<th>FORMATTED VALUE</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Q3b</td>
<td>1950</td>
<td>E</td>
<td>SHOULD BE MISSING</td>
</tr>
<tr>
<td>100</td>
<td>Q4a</td>
<td>1930</td>
<td>E</td>
<td>INVALID VALUE</td>
</tr>
<tr>
<td>200</td>
<td>Q4c</td>
<td>9</td>
<td>I</td>
<td>SHOULD BE MISSING</td>
</tr>
<tr>
<td>300</td>
<td>Q5a</td>
<td>.</td>
<td>M</td>
<td>MISSING VALUE</td>
</tr>
<tr>
<td>400</td>
<td>Q4a</td>
<td>7</td>
<td>E</td>
<td>INVALID VALUE</td>
</tr>
<tr>
<td>600</td>
<td>Q3b</td>
<td>1929</td>
<td>E</td>
<td>SHOULD BE MISSING</td>
</tr>
<tr>
<td>600</td>
<td>Q4a</td>
<td>1999</td>
<td>E</td>
<td>INVALID VALUE</td>
</tr>
</tbody>
</table>