

SAS® Macros used for Editing Survey Data

Elinor Ciftan, Research Triangle Institute, RTP, NC

At RTI we do a lot of statistical analysis of survey data. One of the major tasks we have is to be sure the data are consistent throughout the questionnaire. I have developed a number of SAS macros to help me edit questionnaire data. One example would be "skip pattern editing." An example of a skip pattern would be the question "Have you ever smoked?". The gate question ("Have you ever smoked?") determines whether the respondent answers the detailed questions about smoking usage and patterns. Very often a respondent will answer "no" to the gate question but then answer one of the questions he is supposed to skip. For example, one of the questions may be, "At what age did you start to smoke?". If that question was answered, then the researcher may decide that the respondent should be counted as a smoker. In this example a flag would be set to denote an error and the answer to the gate question would be an "edited" yes. Typically the question will have 1 = yes, 2 = no. An "edited" yes would be given a value of 3.

An example of a macro to determine the presence or absence of a response in a series of questions is as follows:

```
%MACRO ANYHIT(VARS,FLAG);
  &FLAG = 0;
  DO I = 1 TO DIM(&VARS);
    IF &VARS(I) > 0 THEN DO;
      &FLAG = 1;
      I = DIM(&VARS);
    END;
  END;
%MEND ANYHIT;
```

To use this macro the following SAS program segment would create an array of the questions in the skip pattern and then call the macro. Assuming there are four questions about smoking the code would be:

```
ARRAY SMOKING {*} Q2-Q5;
IF Q1 = 2 THEN DO;
  %ANYHIT(SMOKING,FLAG)
  IF FLAG = 1 THEN Q1 = 3;
END;
```

Now we can check for missing answers. We fill in a consistency code whenever a question is not answered. Typically a consistency code for missing data is 8. The macro to accomplish this is as follows:

```
%MACRO PUT_8S(VARS);
  DO I = 1 TO DIM(&VARS);
    IF &VARS(I) = . THEN
      &VARS(I) = 8;
  END;
%MEND PUT_8S;
```

The program code would be:

```
IF Q1 IN (1,3) THEN DO;
  %PUT_8S(QUESTION)
END;
```

To make the macro more flexible, the beginning and ending values of the array can be passed into the macro. Perhaps you have an array for another purpose and you want to deal with a subset of the variables. The macro would look like:

```
%MACRO PUT_0S(VARS,BEG,END);
  DO I = &BEG TO &END;
    IF &VARS(I) = . THEN &VARS(I) = 0;
  END;
%MEND PUT_0S;
```

The following code would insert zeroes into the third through the ninth elements of the QUESTION array:

```
ARRAY QUESTION Q1-Q50;
%PUT_0S(QUESTION,3,9);
```

By using arrays with macros the program becomes much simpler and less prone to error.