A SAS® Macro for Evaluating Whether All, Any, or None of the Variables in a List Have Been Marked - Application for Clinical Data Validation

Art Collins, REGISTRAT, Inc., Lexington, KY

ABSTRACT
For those of us utilizing SAS-based data management systems, a common and repetitive validation task is to evaluate the response to “Mark all that apply” field lists. This is a task which can benefit greatly by application of the SAS® Macro Facility. The macro that will be presented is simple and intuitive to apply to variable lists of all sizes and formats. This is one more tool to add to the tool box which will create efficiencies as well as reduce error. The macro call simply passes the variable list and the value (or values) to be checked for, and returns information about how many of the fields contain the values.

INTRODUCTION
This paper will present four different applications for this macro. The following example from a Cardiovascular study demonstrates a typical application of the macro. The CRF asks the Investigator to indicate whether the patient experienced chest pain, and then to describe the chest pain by “marking all that apply” from a list of descriptors. There might be several queries applying to this question. One might be, if chest pain is marked, but no types of chest pain were indicated. Conversely, there could be types of chest pain indicated, but chest pain was not marked. One way to handle this is to string each variable together in a complex “if and” statement like the following:

```
If CXPAIN='1' or (TYPICAL='1' or REST='1' or EXERTION='1' or NOCTURN='1')
Then EDIT='Chest pain has been listed.';
```

Another situation where you might need to use similar logic is in a “Please mark only one” field. In the example of “Ethnic Group”, not only do we need to make sure that a response has been given, we need to check to see if multiple responses have been given. This query might look like this:

```
If (RACE1='1' and (RACE2='1' or RACE3='1' or RACE4='1' or RACE5='1')) or (RACE2='1' and (RACE1='1' or RACE3='1' or RACE4='1' or RACE5='1')) or (RACE3='1' and (RACE1='1' or RACE2='1' or RACE4='1' or RACE5='1')) or (RACE4='1' and (RACE1='1' or RACE2='1' or RACE3='1' or RACE5='1')) or (RACE5='1' and (RACE1='1' or RACE2='1' or RACE3='1' or RACE4='1'))
Then EDIT='More than one ethnic group was chosen, please choose only the primary group or select "Other" and specify.';
```

A third area where this type of logic is required is when evaluating Inclusion Criteria and all must be answered affirmatively. Below is an example of a query to evaluate inclusion criteria:

```
If (INC1='0' or INC2='0' or INC3='0' or INC4='0' or INC5='0')
Then EDIT='One or more of the inclusion criteria were not met, please confirm the responses.';
```

The final example I would like to discuss is the use of Likert scales. A query of this type of data may be needed to determine if any of the responses were negative. On a seven point Likert scale, a negative response could be any of three values. The query might look like this:

```
If (ITEM1 in ('1', '2', '3') or ITEM2 in ('1', '2', '3') or ITEM3 in ('1', '2', '3') or ITEM4 in ('1', '2', '3') or ITEM5 in ('1', '2', '3'))
Then EDIT='The response to one or more clinical evaluations was negative, please confirm all responses.';
```

The macro that I will present, can be applied to each of these examples, greatly simplifying the code, creating efficiency and reducing the possibility for programmer error.

MACRO SPECIFICATIONS

```
%macro CHECKED(EDT,CHKVAL,FLDS);

length CHECKED EDT CHKVAL FLDS $3.;
array AEDT &FLDS;
ALL='NO';
ATLST1='NO';
NONE='NO';
MULT='NO';
CHECKED=0;
do i=1 to dim(AEDT);
   if AEDT(i) in (&CHKVAL) then do;
      ATMULT=ATMULT+1;
      ATALL=ATALL+1;
      ATNONE=ATNONE+1;
      if AEDT(i) eq dim(AEDT) then ALL='YES';
      if ATMULT eq dim(AEDT) then MULT='YES';
      if ATNONE eq 0 then NONE='YES';
   end;
end;

if CHECKED eq 0 then NONE='YES';
if CHECKED gt 1 then MULT='YES';
if CHECKED eq dim(AEDT) then ALL='YES';
%mend CHECKED;
```

The first parameter, EDT, is a unique identifier for the query that is being parsed. This is necessary because the macro will define an array that is specific to that edit. The second parameter, CHKVAL, is a value or list of values to be checked for. The final parameter, FLDS, is the list of variables that will be checked. The variables will be processed in an array so they must all be of the same type.

The macro returns five pieces of information: whether NONE, ATLST1, MULTiple, or ALL of the fields contain the specified value, and how many of the fields are CHECKED.

APPLICATION

Example 1:

In the first example, “Did the patient experience chest pain? If yes, mark all types that apply”, the data might look like this:

```
CXPAIN TYPICAL REST NOCTURN ATYPICAL EXERTION
0 0 0 0 0 0
0 0 1 0 0 0
```

Then EDT='One or more of the inclusion criteria were not met, please confirm the responses.';
The macro call would be:
\%CHECKED(1,'1',TYPICAL REST NOCTURN ATYPICAL EXERTION);

Once the macro has executed, you can use the information that is returned to parse the queries. The code for the queries becomes:

if CXPAIN='0' and ATLST1='YES' then EDIT='Chest pain is not checked as a Symptom of Ischemia; however, a type of chest pain has been listed.';

if CXPAIN='1' and NONE='YES' then EDIT='Chest pain is checked as a Symptom of Ischemia; however, no type of chest pain has been listed.';

Example 2:
The second example, “Ethnic Group (Mark only one)”, the data might look like this:
RACE1  RACE2  RACE3  RACE4  RACE5
1      0      1      0      0

And the macro call would be:
\%CHECKED(2,'1',RACE1 RACE2 RACE3 RACE4 RACE5);

And the query becomes:
If MULT='YES' then EDIT='More than one ethnic group was chosen, please chose only the primary group or select "Other" and specify.';

In the third example, involving inclusion criteria, there would be a series of questions with yes or no responses. The data might look like this:
INCL1  INCL2  INCL3  INCL4  INCL5  INCL6  INCL7
1      1      0      1      1      1      1

The macro call would be:
\%CHECKED(3,'1',INCL1 INCL2 INCL3 INCL4 INCL5 INCL6 INCL7);

And the query becomes:
If ALL = 'NO' then EDIT='One or more of the inclusion criteria were not met, please confirm the responses.';

Example 3:
In the fourth and final example, the seven point Likert scale, the data might look like this:
ITEM1  ITEM2  ITEM3  ITEM4  ITEM5
7      6      7      3      7

The macro call would be:
\%CHECKED(4,'1' '2' '3',ITEM1 ITEM2 ITEM3 ITEM4 ITEM5);

And the query becomes:
If ATLST1 = 'YES' then EDIT='The response to one or more clinical evaluations was negative, please confirm all responses.';

CONCLUSION
This macro represents both, a tool to increase efficiency and accuracy of validation code, as well as a logical approach to coding several common types of data queries.

CONTACT INFORMATION
Your comments and questions are valued and encouraged.
Contact the author at:
Art Collins
REGISTRAT, Inc.
2525 Harrodsburg Road
Suite 305
Lexington, KY 40504
Phone: 859.223.4334
Fax: 859.223.2005
E-Mail: acollins@registrat.com