A Macro Tool to Check If Values of a Variable Match Its Format

Yuguang Zhao, Sanofi-Synthelabo, Inc., 9 Great Valley Parkway, PO Box 3026, Malvern, PA

ABSTRACT

Formatting is a standard practice for decoding coded variables in SAS. Sometimes, it is necessary to check if the entered values of a coded variable match its format. Unfortunately, it is very tedious to check if the entered values of a variable match its format. This macro will scan all SAS data sets in the specified directory, sort out the formatted variables and compare the entered values with their corresponding formats. If the entered values do not match the values listed in the format, the macro will print out the variable, the entered value and the format name.

INTRODUCTION

In clinical data management, raw data for a number of variables on case report forms are coded, then the codes are entered into the database and code lists are maintained in order to minimize the data entry effort and save resources. Occasionally, it is necessary to check whether the entered values of the coded variable match its format when doing database validations. The typical way of checking would be comparing the frequency of the coded (formatted) variable with the format library. If we want to check a number of or all of the formatted variables in the database, it would be very labor intensive and time consuming.

This macro is developed to improve the database validation process. The macro will scan all SAS data sets in a specified directory and compare the entered values with the format in the format catalog and print out the entered values which do not match the format.

PROGRAM FLOW

Two parameters, LIBNAM and LIBRARY, need to be defined before the macro call. LIBNAM is where the data sets are stored and LIBRARY is where the format catalog is located. If no LIBRARY is defined for the format catalog, the macro will use the default format catalog.

The macro, first, creates a list (FMTCAT) containing all formats in the format catalog with PROC FORMAT with CNTLOUT option. FMTCAT is prefixed with "$" when TYPE="C" so that FMTCAT could be merged with the data set later in the process. A VALUE variable is also created. It takes the value of START when TYPE="C" and the value of LEFT(START) to left align the value when TYPE="N". Then it creates a list for all the data set names in LIBNAM directory with PROC CONTENTS with OUT option followed by sorting with MEMNAME and eliminating duplicates with NODUPKEY. The data set names then are converted to macro variables, DATA1, DATA2, ..., DATAn with CALL SYMPUT() function, where DATA1 corresponds to the first data set, DATA2 to the second data set, DATAn to the last data set and n is the total number of data sets in the directory. The total number of data sets is also put into a macro variable TOTAL.

The macro, secondly, uses a do loop from 1 to TOTAL to go through data set 1 to data set n by feeding the processes with macro variables, DATA1, DATA2, ..., and DATAn. Within the do loop, first, it creates a list containing formatted variables and their corresponding formats followed by converting the variables and formats into macro variables, VAR1, VAR2, ..., VARm and FMT1, FMT2, ..., FMTm where VAR1 and FMT1 correspond to the first formatted variable and its format in the data set, respectively, and m is the total number of formatted variables in the data set. The total number of formatted variables are also put into a macro variable VARTOT. Within the do loop, there are three macros: %GETFREQ, %GETVAR and %SETDATA.

%GETFREQ:

This macro uses a do loop from 1 to VARTOT to go through variable 1 to variable m by feeding procedure with macro VAR1, VAR2, ..., VARm and FMT1, FMT2, ..., FMTm. PROC FREQ with out option produces the frequencies of formatted variables by taking off the corresponding formats and output the result to data sets, FREQ1, FREQ2, ..., and FREQm where FREQ1 is the frequency output for the first variable, FREQ2 is for the second variable and so on for all the variables in the data set.

%GETVAR:

In any of the FREQ1 to FREQm data set, there are three variables, START (renamed the variable name to START as this contains the range start value for the variable), COUNT and PERCENT. In order to merge the FREQ data set with FMTCAT, it is necessary to have variable name and its corresponding format name in the FREQ data sets. With SYMGET() function, it is possible to get VAR1, VAR2, VARm and FMT1, FMT2, ..., FMTm into the FREQ1, FREQ2, ..., FREQm data sets, respectively. Similarly, this is accomplished with a do loop from 1 to VARTOT. A VALUE variable is also created in this macro. If the FMT is a character format, then VALUE takes the value of START. Otherwise, it takes the value of LEFT(START) to left align the entries.

%SETDATA:

%SETDATA sets FREQ1 to FREQm together into one data set FINAL with a do loop.

After %SETDATA macro, FINAL is merged with FMTCAT by FMTCAT and VALUE. The IF condition in the DATA step only keep records exist in the FINAL but FMTCAT data set. Finally, PROC REPORT is used to print the unmatched values.

CONCLUSIONS

The macro presented in this paper will scan all SAS data sets in the specified directory, sort out the formatted variables and compare the entered values with their corresponding formats in the predefined format catalog. If the entered values do not match the values listed in the format, the macro will print out the variable, the entered value and the format name. It is very useful tool to check discrepancies between formatted variables and the format catalog.
ACKNOWLEDGMENTS

I would like to thank my colleague Thomas Guinter for his helpful suggestions.

SAS is a registered trademark or trademark of SAS Institute, Inc. In the USA and other countries.

CONTACTING THE AUTHOR

Yuguang Zhao
Sanofi-Synthelabo, Inc.
9 Great Valley Parkway
Malvern, PA 19355

yuguang.zhao@sanofi-synthelabo.com
ygzhao@bellatlantic.net

APPENDIX

%macro fmtfreq (libnam=);

**********************************************************************;
* Create FMTCAT which contains all formats for the format catalog. *
* The Purpose for FMTCAT is to serve as a master format list for *
* comparing. Every dataset and its formatted variables is checked *
* against FMTCAT in order to find out if the actual values entered *
* have entries in the format catalog.                             *
**********************************************************************;

proc format lib=library cntlout=fmtcat  (keep=fmtname start type);
run;
data fmtcat (keep=fmtname value);
  set fmtcat;
  length value $200.;
  if type='C' then fmtname = '$'||left(fmtname);
  value = start;
  if type='N' then value = left(start);
run;
proc sort data=fmtcat;
  by fmtname value;
run;

**********************************************************************;
* Create DATALIS which contains all the dataset names with format(s) *
* The purpose for DATALIS is to create macro variables with the      *
*  data set names in the directory (LIBNAM).                         *
**********************************************************************;

proc contents data=&libnam.._all_     
  out=datalis(keep=memname format where=(format^=' ')) noprint;
run;
proc sort data=datalis nodupkey;
  by memname;
run;

**********************************************************************;
* Put each dataset name into a macro variables: DATA1 to DATAn,      *
* where n is the total number of data set names in the directory.    *
* Create macro variable TOTAL where TOTAL is the total number of    *
*  data set names in the directory.                                *
* Delete DATALIS to clean up.                                      *
**********************************************************************;

data _null_;            
  set datalis end=eof;   
  call symput ("DATA" ||left(_n_), trim(left(memname)));
  if eof then call symput ("TOTAL", left(_n_));
run;
proc delete data=datalis;
run;
* Macro to get frequency for each variable for each dataset. *;
* - For each dataset, there are FREQ1-FREQm, where m is the total *;
* number of formatted variables in the data set. *;
* - FREQ1-FREQm contain the frequencies of entered value for the *;
* formatted variable *;
**********************************************************************;

%macro getfreq;
proc freq data=&libnam..&&data&j noprint;
  %do i=1 %to &vartot;
    tables &&var&i  /out=freq&i (rename=(&&var&i=start));
    format &&var&i;
  %end;
%mend getfreq;
**********************************************************************;

* Macro to get variable and format name into FREQ1-FREQm, where m *;
*   is the total number of formatted variables in the data set. *;
**********************************************************************;

%macro getvar;
  %do i=1 %to &vartot;
    data freq&i (keep=varname fmtname value count);
    set freq&i;
    length value $200;
    fmtname=symget('FMT'||left(&i));
    varname=symget('VAR'||left(&i));
    %if %substr(&&FMT&i,1,1)=$ %then value=start;
    %else value=left(start); ; /** need the 2nd semicolon to run **/
  %end;
%mend getvar;
**********************************************************************;

* Macro to set FREQ1-FREQm into final for the dataset, where m is *;
*   the total number of formatted variables in the data set. *;
**********************************************************************;

%macro setdata;
  data final;
  set %do i=1 %to &vartot;
    freq&i
  %end; ;                       /** need the second semicolon to run **/
%mend setdata;
**********************************************************************;

* DO loop to go through data set 1 to data set n (DATA1 to DATAn). *;
**********************************************************************;

%do j=1 %to &total;

**********************************************************************;
* Get variable name and format name from each dataset: *;
* VARTOT1-VARTOTn where n is the total number of data set in the *;
* directory. *;
* The variable name and format in VARTOT1-VARTOTn is put into macro *;
* variables: VAR1-VARM and FMT1-FMTm, where m is the total number *;
* of formatted variables in the data set (any of DATA1-DATAN). *;
* VARTOT is the total number of formatted variables in the data set.*;
* Delete VARTOT1-VARTOTn to clean up. *;
**********************************************************************;

proc contents data=&libnam..&&data&j
  out =varfmt&j (keep=name format where=(format^in (' ', 'DATE', 'TIME')))
    memtype=data noprint;
run;
%let vartot=0; /* default value if there is no formatted variable in the dataset */
data _null_;  
set varfmt&j end=endof;  
call symput('VAR'||left(_n_),name);  
call symput('FMT'||left(_n_),format);  
if endof then call symput('VARTOT',left(_n_));  
run;
proc delete data=varfmt&j;  
run;
%if &vartot ^= 0 %then %do;
**********************************************************************;
* Get actual value frequency for each variable from each dataset  *
**********************************************************************;
%getfreq;
run;
**********************************************************************;
* Get variable and format name into frequency output datasets  *
* so that to merge them with fmtcat dataset.  *
**********************************************************************;
%getvar;
run;
**********************************************************************;
* Set the frequency datasets together.  *
* The data set now contains the variable name, the format and the  *
* actual value  *
**********************************************************************;
%setdata;
run;
%end;
proc sort data=final;
  by fmtname value;
run;
**********************************************************************;
* Merge with format library dataset FMTCAT.  *
* Only print the value that does not match formats.  *
* Excluding date and time formats.  *
**********************************************************************;
data final;
  merge fmtcat  (in=a)  
    final  (in=b);
  by fmtname value;
  if b and ^a ;
run;
proc sort data=final;
  by varname fmtname value;
run;

  title  "Printout of &libnam dataset &data&j";
  title2 "Actual value does not match the formatted value";
  proc report data=final headline headskip split='**' nowindows;
  column varname fmtname value count;
  define varname / width = 20 'Variable*Name';
  define fmtname / width = 20 'Format*Name';
  define value / width = 20 'Actual*Value';
  define count / width = 15 center 'Number of*Observations';
run;
%end;
%mend fmtfreq;