

%FREQ1VAR: Frequency of one variable with format: a macro to standardize proc FREQ output data sets

Ronald Fehd, Centers for Disease Control and Prevention, Atlanta GA

ABSTRACT

SAS(r) software provides the ability to associate formats, or look up tables, with variables in a data set. Encapsulating the format labels in a summary data set provides a complete and independent set of information about a variable to other procedures.

This macro was written to standardize summary data sets and bring them into conformance with data sets produced by Fehd's (1997) CHECKALL and SHOWCOMB macros, which provide information about multiple response data. The standardized data sets, or objects, have the same structure and are used for quick and concise reporting of summary information of large survey questionnaire data sets. Their identical structure enables easy access by other methods and software.

INTRODUCTION

Proc FREQ provides a summary of a variable; when a format is associated with the variable the procedure prints the value using its associated format. While a paper report is often sufficient, the increasing demand for graphics reports and presentations creates an expectation for an object that contains a complete set of information about a data set and the variable(s) being reported.

This macro processes a proc FREQ output data set and adds format labels and attributes to the data set. An attribute is here defined as metadata, that is, data about the data, which may not be organized as rows and columns of a data set, but exists as a single item of information. These attributes include the name of the data set against which the proc was run, the number of observations of that data set, the number of valid responses in the variable and the response rate: the ratio of valid response to number of observations. These attributes are placed in variable labels and the first observation of the summary data set.

Setup: Sets of Values, AutoExec, proc FORMAT

Observations in a data set contain one of three sets of values. When a variable is defined, its value is initialized to blank for character variables, or missing for numeric variables. As data entry or data manipulation occurs, variables receive valid, or expected, values. In a small minority of cases, variables obtain invalid values.

During the data review process, it is necessary to report all three sets of data values, but for summary purposes, blank/missing and invalid values are expected to be excluded. This is accomplished in this macro by using standardized labels in proc FORMAT value statements. Character variables have space and dot labeled as "&BLANK.". Numeric variables have missing and in some cases, zero, labeled as "&BLANK.". Valid values have their description. Values to be excluded are grouped with the phrase: other="&INVALID.".

The standard labels for blank/missing and invalid are set as global macro variables in the autoexec for the session. This allows the labels to be assigned with proc FORMAT in a separate program from the summarization program.

Refer to the test data with the macro for examples.

Three steps: proc, read and calculate, save to library

This macro consists of three main steps: using proc FREQ to produce an output data set; reading the data to prepare the attributes and writing the summary data set to the library,

Proc FREQ is the first step in the macro. Data are excluded using a where statement which compares the formatted value of the variable to the two macro variables: BLANK and INVALID. Parameters are provided to sort the output data set by descending Count and to subset the data. If the output data set contains no observations, then the macro stops processing.

Step 1 uses the %NOBS macro to provide the number of observations of the data sets. The %NOBS macro is based on the %OBSNVARS macro (*SAS Macro Language Reference, First Edition*).

In the second step, the format labels are added to the data set. The width of the three major variables: Label, Count and Percent, are calculated. The number of responses is accumulated from Count. The label of the tables variable is copied to a macro variable; other macro variables are created containing the various widths and the number of responses. When the data step is complete then the percentage of response is calculated.

In the final data step, the attributes are placed in their respective variable labels as the summary data set is written to the library.

CONCLUSION

Proc FREQ provides an output data set that contains three variables: the tables variable, plus Count and Percent. This macro renames the tables variable, adds a character variable with the format label, calculates and adds other attributes to the data set to create a complete package of information in a standardized object that can be used by other methods and applications for graphics and presentations.

REFERENCES

- Fehd, Ronald (1997), "%CHECKALL, a macro to produce a frequency of response data set from multiple-response data," *Proceedings of the Twenty-Second Annual SAS Users Group International*, 22:1084.
- Fehd, Ronald (1997), "%SHOWCOMB: a macro to produce a data set with frequency of combinations of responses from multiple-response data," *Proceedings of the Twenty-Second Annual SAS Users Group International*, 22:939.
- SAS Institute Inc. (1997), *SAS® Macro Language Reference, First Edition*, Cary, NC: SAS Institute Inc.

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

Author: Ronald Fehd **e-mail:** RJF2@cdc.gov
Centers for Disease Control **MS-G25**
4770 Buford Hwy NE
Atlanta GA 30341-3724 **voice:** 770/488-8102

SAS-L archives: send e-mail
to: SAScontrib@SASServ.uga.edu
for %FREQIVAR subject: cntb0038: download
for %CHECKALL subject: cntb0032: download
for %SHOWCOMB subject: cntb0033: download

ACKNOWLEDGMENTS

This routine was developed over a period of ten years while I

crunched the numbers of survey data collected by the Model Performance Evaluation Program (MPEP) of the Division of Laboratory Systems, Public Health Practice Program Office of the Centers for Disease Control and Prevention, Atlanta, Georgia. John Hancock, chief, Information Services Activity, encouraged me to write up these routines. I wish to thank Sharon Blumer, David Cross, Thomas Hearn, and William Schalla of the MPEP group for their perseverance while I developed this routine.

```

%MACRO NOBS(_MAC_VAR,DATA=.,_GLOBAL=0)
/STORE DES = 'returns mac-var w/NOBS: SYSLAST or DATA' /* */
;
/*-----*/
RUN;
%IF "&_MAC_VAR" = "" %THEN %LET _MAC_VAR = NOBS;
%IF &_GLOBAL %THEN %DO; %GLOBAL &_MAC_VAR; %END;
%IF &DATA = . %THEN %LET DSN = SYSLAST; /* resolve mac-vars in name;
%ELSE %LET DSN = &DATA;
%LET DSID = %sysfunc(open(&DSN));
%IF &DSID %THEN %DO;
%LET &_MAC_VAR = %sysfunc(attrn(&DSID,NOBS));
%LET N_VARS = %sysfunc(attrn(&DSID,N_VARS));
%PUT NOBS returns nvars=&N_VARS; /*&N_VARS;
%LET RC = %sysfunc(close(&DSID)); %END;
%ELSE %PUT Open for data set &DATA. failed - %sysfunc(sysmsg());
%PUT NOBS: '&_MAC_VAR'=<&&_MAC_VAR> data=&DSN;
/*-----*/
%NOBS*/%MEND;

/*-----*/
* MACRO: FREQIVAR returns summary data set
* from data set, variable and format
* AUTOEXEC: %LET BLANK =BLANK;
* %LET INVALID=INVALID;
* %LET DATA_SET=<data-set-name>;
* USAGE: 1) %FREQIVAR(var-name,format);
* 2) %FREQIVAR(var-name,format,out=ABC);
* 3) %FREQIVAR(var-name,format,PRINT=1);
* 4) %FREQIVAR(var-name,format,TESTING=1);
* 5) %FREQIVAR(var-name,format,GRFXPATT=PIE);
* 6) %FREQIVAR(var-name,format,ORDER=DATA);
* 7) %FREQIVAR(var-name,format,WHERE=Var1 eq 'A');
* do not use comparison operator symbols: =
* use comparison operator mnemonics: eq
* PROCESS:
* 1. proc FREQ
* %order=freq?
* apply format
* where: exclusion of BLANK, INVALID
* other subset if present
* %NOBS: if output data set is empty then exit
* 2. read data
* make label using format
* if Colon in Label then front-trim Label
* if grfxpatt = PIE then append Count=Percent to Label
* calculate max widths, accumulate Count to MbrResp
* make mac-vars: VarLabel MbrResp
* max-width of: Label, Count, Percent
* calculate %Resp
* 3. make output data set
* rename: name(Variable) := in[ValueChr, ValueNum]
* place attributes in variable labels
* NOTES:
* * mac-vars BLANK and INVALID usually defined in autoexec
* * so format program and macro can access them as global variables
* * variable labels may have colons, if so, front-trim to colon
* * e.g.: '006: Supervisor'; change to: 'Supervisor'
* KEYWORDS: autoexec formats %NOBS FREQ object standardization
* AUTHOR: Ronald Fehd e-mail: RJF2@cdc.gov
* Centers for Disease Control, and Prevention
* 4770 Buford Hwy NE MS-025 fax: 770/488-8282
* Atlanta GA 30341-3724 voice: 770/488-8102
/*-----*/
%MACRO FREQIVAR(/*-----*/
VARIABLE /* variable name */
,FORMAT /* format of variable with suffix=dot */
,DATA =&DATA_SET /* DATA_SET is global variable, else hardcode here */
,BLANK =&BLANK /* common format label indicating blank/missing */
,INVALID=&INVALID /* common format label indicating out of range */
,GRFXPATT=&GRFXPATT /* graphics-pattern in (barv barv pie) */
,LIBRARY=WORK /* library name for read and write, LIBRARY=LIBRARY */
,LBL_VAR =. /* label of variable, see %LOCAL VARLABEL */
/* provide to overwrite &DATA_SET label of &VARIABLE */
,LBL_CNT =Number of Laboratories Responding/*
/* label of frequency count */
,LBL_PCT =Percentage of Laboratories Responding/*
/* label of frequency percent */
,OUT =. /* output data set name, if not &VARIABLE. */
,PRINT =0 /* ?print output data set?, PRINT=1 */
,ORDER=FREQ /* ?sort descending Count?, else ORDER=DATA */
,WHERE =. /* where statement for subsetting */

```

```

/* where=var1 eq 'A' */
/* where=var1 eq 'B' and var2 eq 'A' */
/* where=var1 eq 'B' and not var2 eq 'A' */
,TITLE =FREQIVAR/* title for graphics */
,TESTING =1/* ?print intermediate data-sets and messages? ,TESTING=0 */
);%IF &TESTING %THEN %DO; options mprint notes; %END;
%ELSE %DO; options noprint nonotes; %END;

%*1;proc FREQ data = &LIBRARY..&DATA.
%IF &ORDER=FREQ %THEN order = freq; %proc closure:;
format &VARIABLE. &FORMAT.;
tables &VARIABLE. / out = FREQIVAR noprint;
where put(&VARIABLE.,&FORMAT.) not in("&BLANK","&INVALID")
%IF "&WHERE" ne "" %THEN and &WHERE.; %where closure:;

%IF &TESTING %THEN %DO;proc PRINT data = FREQIVAR; %END;
%local NOBS;%NOBS(NOBS);run;
%IF not &NOBS %THEN %DO;%PUT %FREQIVAR: obs=0 for &VARIABLE. &FORMAT.;
%GO TO ENDDMAC; %END;
%NOBS(NOBS,DATA=&LIBRARY..&DATA.);run;
%local LENGTH;%LET LENGTH = %length(&NOBS);

%*2.1;DATA FREQIVAR;
length Label VarLabel $ 40;
%local WLabel WCount WPercent MbrResp VarLabel;
retain WLabel WCount WPercent MbrResp 0;
drop WLabel WCount WPercent MbrResp VarLabel Colon;
do until(EndOfFile);%*-----
set FREQIVAR end = EndOfFile;
Label = left(put(&VARIABLE,&FORMAT.));
%if colon present, front-trim to colon;
Colon = index(Label,':');
if Colon then Label = left(substr(Label,Colon+1));
%if %index(%upcase(&GRFXPATT.),PIE) %THEN %DO;
%append "new-line" Count (Percent)" to Label;
Label = trim(Label) || ' ' || trim(left(put(Count ,&LENGTH..0)))
|| ' ' || trim(left(put(Percent,5.1)))
|| '%'; %END;
WLabel = max(WLabel ,length( Label ));
WCount = max(WCount ,length(trim(left(put(Count ,&LENGTH..0)))));
WPercent= max(WPercent,length(trim(left(put(Percent ,5.1 )))));
MbrResp = Count;
output; %*-----do until(EndOfFile); end;

%*2.2 make mac-vars;
call label(&VARIABLE.,VarLabel);
%change quote to explanation mark to avoid mac-error;
VarLabel = translate(VarLabel,',' ,');
%if colon present, front-trim to colon;
Colon = index(VarLabel,':');
if Colon then VarLabel = left(substr(VarLabel,Colon+1));
call symput('VARLABEL',trim(left( VarLabel )));
call symput('MbrRESP',trim(left(trim(MbrResp,&LENGTH..0))));
call symput('WLabel',trim(left(put(WLabel ,2. ))));
call symput('WCount',trim(left(put(WCount ,&LENGTH..0))));
call symput('WPERCENT',trim(left(put(WPercent,5. )))); stop;run;

%LET PCHTRESP = %val(100* &MbrRESP /&NOBS);
%IF &TESTING %THEN %DO;proc PRINT data = FREQIVAR;
%put PCHTRESP=<&PCHTRESP> WLabel=<&WLabel>; %END;

%*3;%IF "&OUT." eq "" %THEN %LET OUT = &VARIABLE.;
DATA &LIBRARY..&OUT.(label = "FREQIVAR: &VARIABLE. fmt: &FORMAT."
rename=<&VARIABLE =
%IF "%substr(&FORMAT.,1,1)" = "%S" %THEN ValueChr;
%ELSE ValueNum; %DATA closure:;));
attrib %local LENGTH;
%LET LENGTH = %val(2 + %length(&MbrRESP.));
N_eq length = $ &LENGTH.
format = $char&LENGTH..
Label = "N=&MbrRESP data:&DATA Obs:&NOBS Resp:&PCHTRESP.%";
%IF "&LBL_VAR" = . %THEN %LET LBL_VAR = &VARIABLE.;
Label length = $ &WLabel. label = "&LBL_VAR."
format = $char&WLabel..
Count length = 4 label = "&LBL_CNT."
format = &WCount..0
Percent format = &WPERCENT..1 label = "&LBL_PCT."
&VARIABLE label = "&VARIABLE. value"
GrfxPatt length = $ 8 label = "&grfxpatt: &GRFXPATT"
format = $char8.
%IF "&WHERE" ne "" %THEN %DO;

```

```

Subset length = $ 6 label = "WHERE."
format = $char. %END;
%LET LEN =length(ATITLE.);
Title length = $ ALLEN. label = "Title"
format = $char$LEN.;
retain Title "ATITLE." N_eq "N=NUMBERRESP." GrfxPatt "&GRFXPATT"
%IF "WHERE" ne "." %THEN Subset 'subset';
do until(EndOfFile);
set FREQIVAR end = EndOfFile;
output; Title='.';N_eq='.';GrfxPatt='.';
%IF "WHERE" ne "." %THEN %DO; Subset='.'; %END;
%*.....do until(EndOfFile)*; end;
stop;

%*4; %IF %PRINT or %TESTING %THEN %DO;
proc PRINT data = $LIBRARY..$OUT. double label nobds;
format _all_;
proc CONTENTS data = $LIBRARY..$OUT.; %END;
%ENDOMAC:RUN; %*.....* %END;
/*test data.....*/
%*autotexes;%LET DATA_SET=TESTDATA;%LET BLANK=BLANK;%LET INVALID=INVALID;
%libname LIBRARY <lib-ref>;
proc FORMAT;
value $C1_4_ '1'='C1-4:one-North'
'2'='C1-4:two-East'
'3'='C1-4:three-South'
'4'='C1-4:four-West' ' ' '.'='&BLANK.' other='&INVALID.';
value $CA_C_ 'A'='CA-C:apple'

```

```

'B'='CA-C:banana'
'C'='CA-C:cherry' ' ' '.'='&BLANK.' other='&INVALID.';
value N1_3_ 1 = 'N1-3: one'
2 = 'N1-3: two'
3 = 'N1-3: three' ' ' '.'='&BLANK.' other='&INVALID.';
value N1_9_ 1 - 4 = 'N1-9: 1..4'
5 - 9 = 'N1-9: 5..9'
9 - high = 'N1-9: >9' ' ' '.'='&BLANK.' other='&INVALID.';
data TESTDATA;
attrib C1_4 length=$1 format=$C1_4_ label='Chr four directions'
CA_C length=$1 format=$CA_C_ label='Chr fruit list'
N1_3 length= 4 format= N1_3_ label='Num three levels'
N1_9 length= 4 format= N1_9_ label='Num three ranges'
;do N1_9 = 0 to 9; N1_3 = N1_9; C1_4 = put(N1_9,1.);
if N1_9 then CA_C = substr('ABCDEFGH',N1_9,1);
if mod(N1_9,2) then output; output; end; stop;
proc PRINT data = TESTDATA;format _all_;proc CONTENTS data = TESTDATA;
/*****
%FREQIVAR(N1_3, N1_3.);
%FREQIVAR(N1_9, N1_9.);
%FREQIVAR(C1_4, $C1_4.);
%FREQIVAR(CA_C, $CA_C.);
%FREQIVAR(N1_3, N1_3_, GRFXPATT=FILE);
%FREQIVAR(N1_3, N1_3_, ORDER=DATA);%*descending Count;
%FREQIVAR(N1_3, N1_3_, WHERE=N1_3 gt 3);%*obs=0 msg;
/* END TEST SECTION .....*/

```