

%CHECKALL: a macro to produce a frequency of response data set from multiple-response data

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

ABSTRACT

Multiple-response data from survey questionnaires where questions have the instruction "check all that apply" present a challenge to the SAS® software programmer. For a simple question, the answer may be either A or B; the sum of percent response is 100%. For the series of variables in a multiple-response question, the answer may be both A and B; because response rate for each variable in the series is dependent upon the other variables, the sum of the percent response may be greater than 100%.

This paper examines the SAS software proc FREQ output data set and discusses the construction of a standardized data set containing frequency of response information for multiple-response data for graphics presentation.

The CHECKALL macro takes as parameters an output data set name, and a list of the multiple-response variables. The output data set contains the variable names and labels of the multiple-response variables. This data set is used by the SHOWCOMB macro to report the frequency of combinations of response. See Fehd (1996), (1997), %SHOWCOMB and %ARRAY.

INTRODUCTION

Simple questions may have simple answers. Simple questions may have complex answers when the question contains the phrase "Check all that apply". This paper reviews the output data set of proc FREQ on a single variable with a format. This data set is used as a model to construct a macro which produces a standardized data set with the frequencies of the each variable in the series of multiple-response data.

The Answer: (Choose only One)

Programmers may or may not be consulted when survey questionnaires are designed. When data are delivered we hope that the answers are in this form:

A:Apple B:Banana C:Cherry >-> box.for.answer(_)

When data are simple, the programming exercise is straightforward. SAS software provides proc FORMAT to translate variable values into information. Proc FREQ can create an output data set, which can be standardized by adding the format as a variable when saved for later use.

Program 1: Choose Only One

```
data CHZONLY1; input Q01 $ 1; cards;

proc FORMAT; value $Q01 'A' = 'Apple'
                    'B' = 'Banana'
                    'C' = 'Cherry';

proc FREQ data = CHZONLY1;
  tables Q01 / noprint out = FREQ;

data Q01; length Format $ 40;
  set FREQ; Format = Put(Q01,$Q01.);

proc PRINT data = Q01; sum _numeric_;
```

```
SAS output:
Q01 Format Count Percent
--- -----
A Apple 1 16.6
B Banana 2 33.3
C Cherry 3 50.0
--- -----
6 100.0
```

This is the data set that we will use as a model for the information in a multiple-response series. It contains the information Value (Q01), Label (Format), Count, and Percent of one variable. The internal relationships of this data are that the sum of Count is equal to the number of responses and the sum of Percent is 100%. These internal relations may not hold for multiple-response data.

The Answers: (Check all that apply)

```
A:Apple >----> box.for.answer(_)  
B:Banana >----> box.for.answer(_)  
C:Cherry >----> box.for.answer(_)
```

What values mean "Checked"? The most common sets include: (Y,N), (T,F), (X,<dot>), etc. Here we use (1,0).

The next example data set illustrates multiple responses. The variables have labels, which will be used in place of the format. We use the same proc FREQ output option and examine the result.

Program 2: Check All

```
data QUERIES; label Q02A = 'Apple'
                  Q02B = 'Banana'
                  Q02C = 'Cherry';
  input Q02A Q02B Q02C; cards; ;

proc FREQ data = QUERIES;
```

```
tables Q02A / noprint out = Q02A;
tables Q02B / noprint out = Q02B;
tables Q02C / noprint out = Q02C;
```

```
DATA Q02; set Q02A Q02B Q02C;
```

```
proc PRINT data = Q02;
```

SAS output:

Q02A	Q02B	Q02C	Count	Percent
0	.	.	1	20.0
1	.	.	4	80.0
.	0	.	4	80.0
.	1	.	1	20.0
.	.	0	3	60.0
.	.	1	2	40.0

This data set has some of the desired information. We will eliminate the observations with values of zero, which represent 'not checked', combine the series of variables into one variable and save the label. This process will be repeated for each of the individual data sets of the series.

Program 3: Standardize FREQ data set

```
data Q02A;
drop Q02A;
length
  Value $ 8
  Label $ 40;
retain Value 'Q02A';
set Q02A;
where Q02A = 1;
call label( Q02A,Label);
```

```
*repeat process for Q02B, Q02C;
```

```
DATA Q02; set Q02A Q02B Q02C;
```

```
proc PRINT data = Q02; sum _numeric_;
```

SAS output:

Value	Label	Count	Percent
Q02A	Apple	4	80.6
Q02B	Banana	1	20.0
Q02C	Cherry	2	40.0
.	.	-	-----
.	.	7	140.0

This is the desired product: a data set, or object, with a known set of variable names where each observation contains the name of the multiple-response variable, its label, the count and percent response. The sum of Count is greater than the number of observations; the sum of Percent is greater than 100%. The internal consistency of a uniquely answered question is apparently lost. This information can be saved elsewhere.

Parameters for the macro

Our task now is to replicate this process for any series. The minimum parameter would be the name of the series of variables. In the test data presented here that name is the prefix of each of the series' variables. This parameter is used as the output data set name. For one-time usage, a list of variables would be the second parameter. In production usage, data sets with the series of variable names are prepared from subsets of a proc CONTENTS output data set. The data set names are in the form: V&SERIES. See the test data provided with the macro below. In order to process each variable in the series, the list is converted to a series of macro variables with the macro %ARRAY routine, which accepts either a list or a data set as input. See Fehd (1996) %ARRAY. The default list for CHECKALL will be a data set previously prepared, with the option of accepting a user-prepared list.

Macro step 1:

```
%IF "&LIST"="DATA" %THEN
%ARRAY(VAR,DATA=&LIBRARY.V&SERIES.,VAR=Name);
%ELSE %ARRAY(VAR,&LIST.);
```

The macro call %ARRAY(VAR,&LIST) performs the following macro variable assignments:

```
%LET VAR1 = Q02A;
%LET VAR2 = Q02B;
%LET VAR3 = Q02C;
%LET DIM_VAR = 3;
```

If the LIST is provided in a data set, then the macro variables must be concatenated in order to be used in the SAS language keep and array statements. The parameter LIST has a dummy value; it is used to contain the concatenated list.

Macro step 2:

```
%MACRO VAR_LIST;
%DO I = 1 %TO &DIM_VAR; &&VAR&I. %END; %MEND;
%IF "&LIST"="DATA" %THEN %LET LIST = %VAR_LIST;
```

Data set information and subsetting, TRUE

Since the internal consistency of these data sets is known to be unstable, then both the number of observations and response rate must be saved to be stored with the data set. An observation is output when one or more of the variables is 'checked'. In order to accommodate different sets of data values, a parameter TRUE is provided. The TRUE parameter allows values to be picked from multi-valued data such as ('A','B','C'..). For the next example program the value of TRUE is numeric one. The test data provided with the macro contains a series of variables where values of 'checked' are equal to character one, numeric one, and another series with 'X' meaning 'checked'.

Macro step 3:

```

* macro parameter: TRUE=1
DATA CHECKALL;
  retain NnbrChkd NnbrResp 0;
  set LIBRARY.&DATASET
    nobs = NnbrObs ;
  NnbrChkd = 0;
  array CheckAll {*} &LIST.;
  do I = 1 to dim(CheckAll) until(NnbrChkd);
    NnbrChkd + (CheckAll{I}= &TRUE.);    end;
  if NnbrChkd then do; output;
                                NnbrResp + 1;    end;

call
symput("NMBROBS",trim(left(put(NnbrObs,8)))));
call
symput("NMBRRESP",trim(left(put(NnbrResp,8)))));
run;
%LET PCNTRESP = %eval(100* &NMBRRESP /&NMBROBS);

```

Loop through the series of variables

The macro %ARRAY returned the dimension of the series of variables to be processed in the variable DIM_VAR. This allows the following loop definition; each variable is accessed by the convention &&VAR&I., where VAR is the name of the macro array and I is the index.

```

%DO I = 1 %TO &DIM_VAR;
  *... (keep = &&VAR&I.); %END;

```

As in Program 2 above, each variable is run through proc FREQ and output to a data set. This data set is standardized and concatenated to a work data set for the next step.

Macro step 4:

```

%DO I = 1 %TO &DIM_VAR; /*-----*/
proc FREQ data = CHECKALL (keep = &&VAR&I.);
  tables    &&VAR&I. / noprint out = FREQ;

DATA FREQ; length Value $ 8
              Label $ 40;
  retain Value "&&VAR&I.";
  set FREQ;
  where    &&VAR&I. = &TRUE;
  call label( &&VAR&I,Label);

%IF &I = 1 %THEN %DO;
  DATA SERIES; set FREQ; %END;
%ELSE %DO;
  proc APPEND base = SERIES data = FREQ; %END;
  /*..... %DO I=1:&DIM_VAR*/ %END;

```

Optimizing for storage

While the data sets created by this routine have the same set of variables, it is necessary for later processing to optimize their storage length. This data step checks the maximum width of the Label. In the last data step these values will be used to change the width of Label.

Macro step 6:

```

DATA _NULL_;
  retain WLabel 0 ;
  set SERIES;
  WLabel = max(WLabel, length(Label));
  call
symput('WLABEL',trim(left(put(WLABEL,2)))));

```

Save the data for later usage

As the data is saved to a permanent data set the width and format of Label are added.

Macro step 7:

```

DATA LIBRARY.&SERIES.;
  attrib Label length = $    &WLABEL.
              format = $char&WLABEL.;
  set SERIES;

```

CONCLUSION

When analyzing a single variable with unique responses procedures FREQ and FORMAT can be combined to produce a data set containing information about that variable. When given a multiple-response data set to analyze, correct labels need to be applied to the variables in the order to be able to produce a data set which describes the series in the same fashion as a single variable. For similar processing of combinations of responses to multiple-response data see Fehd, (1996), %SHOWCOMB.

REFERENCES

Fehd, Ronald (1996), " %ARRAY, construction and usage of arrays of macro variables" Proceedings of the Fourth Annual Conference of the SouthEast SAS Users Group, 156-160.

Fehd, Ronald (1996), "%SHOWCOMB: a macro to produce a data set with frequency of combinations of responses from multiple-response data." Proceedings of the Fourth Annual Conference of the SouthEast SAS Users Group, 399-402.

Fehd, Ronald (1997) %ARRAY, %CHECKALL, %SHOWCOMB: Proceedings of the Twenty-Second Annual SAS Users Group International Conference.

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

Author: Ronald Fehd
Centers for Disease Control
4770 Buford Hwy NE MS-G25
Atlanta GA 30341-3724 voice: 770/488-4316
e-mail: RJF2@phpdLs1.em.cdc.gov (D eL S one)
SAS-L archives: send e-mail

to: SAScontrib@SASServ.uga.edu
 for %ARRAY subject: cntb0031: download
 for %CHECKALL subject: cntb0032: download
 for %SHOWCOMB subject: cntb0033: download

ACKNOWLEDGMENTS

These routines were developed over a period of five 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 repeating the questions about the data often enough and in enough variation that I was finally able to comprehend the underlying pattern.

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

```

/* *****
* MACRO: CHECKALL
* NOTE: uses macro ARRAY
*
* USAGE: 1) %CHECKALL(series-name);*assumes a data set contains vars
* 2) %CHECKALL(series-name,LIST=var1 var2 var3 .. varN);
* 3) %CHECKALL(series-name,DATA=dataset);
* 4) %CHECKALL(series-name,TRIMCHAR=-);
* 5) %CHECKALL(series-name,PRINT=1);
* 6) %CHECKALL(series-name,SORT=0);
* 7) %CHECKALL(series-name,BY_VAR=var-name);
* 8) %CHECKALL(series-name,BY_VALUE=var-name);
* 9) %CHECKALL(series-name,BY_VALUE='B');
* 10) %CHECKALL(series-name,TRUE='Y');
*
* DESCRIPTION:
* Processing of a series of multiple-response variables,
* usually identified by instruction "Check-all-that-apply",
* to produce a standardized output dataset
* containing frequency of each variable
* vars: Value Label Count Percent
* to be used as input to graphics routines
*
* PROCESS:
* 1: macro: prepare macro ARRAY of variables
* 2: macro: if LIST=DATA then concatenate elements into LIST
* 3: data: prepare subset of DATA
* 4: macro loop: for each variable
* 5: proc: if wanted SORT output dataset
* 6: data: save various variable widths
* 7: data: save optimized dataset
* 8: proc: if wanted PRINT output dataset
*
* NOTES:
* LIST=DATA: default is data set previously prepared
* from proc CONTENTS output data set
* assumes all variables are labeled
* assumes macro ARRAY is available
* picks variables with value = &TRUE,
* can be used with other than binary-valued data: (0,1) ('0','1')
* will pick single value from multi-valued data: ('A','B','C'..)
*
* KEYWORDS: APPEND array %ARRAY call label dim() dimension FREQ left
* multiple-response data put trim
*
* Author: Ronald Fehd
* Centers for Disease Control PHPPD DLS
* 4770 Buford Hwy NE NS-G25 fax: 404/488-7667
* Atlanta GA 30341-3724 voice: 404/488-4316
* e-mail: RJF2phpd@sls1.em.cdc.gov (P H P D o L S one)
*
* *****
* MACRO CHECKALL( /*-----*/
* SERIES /* name of series of check-all variables */
* ,LIST=DATA /* output dataset name */
* /* list of variables,
* /* default is DATA previously prepared
* /* whose name is V&SERIES
* ,LIBRARY=LIBRARY /* library name
* ,DATA=&DATASET /* DATASET is global variable, else hardcode here
* /*-----*/
* /* DO NOT USE &SERIES AS NAME OF THIS PARM
* /*-----*/
* ,LBL_LBL=&SERIES Label/*Label of variable Label
* ,LBLCOUNT=Number of Laboratories Responding/*
* /* Label of Freq-Count
* ,LBLPCENT=Percentage of Laboratories Responding/*
* /* Label of Freq-Percent
* ,BY_VAR= /* var for subsetting
* ,BY_VALUE='A' /* var for subsetting: value
* ,CHARTYPE=QOQBARN /* Chart-Type in (QOQBARN QOQBARV)
* ,PRINT=1 /* ?print output dataset? used while testing
* ,TESTING=0 /* ?enable testing features?
* ,SORT=1 /* ?output dataset to be sorted by descending Count?
* ,TITLE=TEMP /* Label of output dataset and title for graphics
* /* TITLE cannot contain commas
  
```

```

,TRIMCHAR= /* Front-trim label to this char
/* Labels expected to be in form:
/* 'Q04B: Category - Specific'
/* ,TRUE= 1 /* variable counted with this value: numeric one
/* ,TRUE='1' /* variable counted with this value: character one
/* ,TRUE='Y' /* variable counted with this value: character 'Yes'
);/*-----*/
%IF &PRINT
or &TESTING %THEN %DO; options mprint notes; %END;
%ELSE %DO; options noprint nonotes; %END;

%*1: macro: prepare ARRAY of variables
%*data is subset from proc CONTENTS: Var-Name is in variable Name;
%*IF 'LIST'='DATA' %THEN %ARRAY(VAR,DATA=&LIBRARY..V&SERIES.,VAR=Name);
%*ELSE %ARRAY(VAR,&LIST.);

%*2: macro: if LIST=DATA then concatenate elements into LIST;
%*MACRO VAR_LIST;%DO I = 1 %TO &DIM_VAR; &&VAR&I. %END; %*END;
%*IF 'LIST'='DATA' %THEN %LET LIST = %VAR_LIST;

%*3: data: prepare subset of DATA: delete obs w/no responses;
DATA CHECKALL;
retain NnbrChkd NnbrResp 0;
do until(EndOfFile);
set &LIBRARY..&DATA
(keep = &BY_VAR. &LIST.)
end = EndOfFile;
nobs = NnbrObs;
NnbrChkd = 0;
array CheckAll {*} &LIST.;
drop I;
do I = 1 to dim(CheckAll)
until(NnbrChkd); NnbrChkd +
(CheckAll{I} = &TRUE.); end;

if NnbrChkd
%IF &BY_VAR ne %THEN %DO;
and &BY_VALUE. %END;
then do; output;
NnbrResp + 1; /*if NnbrChkd*/ end;
/*do until(EndOfFile)*/ end;
call symput('NMBROBS',trim(left(put(NnbrObs,8)))));
call symput('NMBRRESP',trim(left(put(NnbrResp,8)))));
stop; run;
%LET PCHTRESP = %eval(100 * NMBRRESP / NMBROBS);
%put PCHTRESP = <&PCHTRESP.>;
**+;
%IF &TESTING %THEN %LET DIM_VAR = 2; *Q04B:10;

%*4: macro loop: for each variable: FREQ, standardize data, APPEND;
%local I;
%DO I = 1 %TO &DIM_VAR; /*-----*/

proc FREQ
data = CHECKALL (keep = &&VAR&I. &BY_VAR.);
tables &&VAR&I. / noprint out = FREQ;
%IF &BY_VAR ne %THEN %DO;
by &BY_VAR.; %END;

DATA FREQ;
drop &&VAR&I.;
length Value $ 8
Label $ 40;
retain Value &&VAR&I.;
do until(EndOfFile);
set FREQ end = EndOfFile;
where &&VAR&I. = &TRUE;
call label(&&VAR&I,Label);
if index(Label,&TRIMCHAR) then Label = left(
substr(Label,
index( Label,
&TRIMCHAR)+1));
output; /*do until(EndOfFile)*/ end;
stop;

%IF &I = 1 %THEN %DO;
DATA SERIES;
  
```

```

do until(EndoFile);
  set FREQ end = EndoFile;
  output;
stop;
%ELSE %DO;
proc APPEND base = SERIES
  data = FREQ;
/*..... %DO I=1:&DIX_VAR*/ %END;

%5: proc: if wanted SORT output dataset;
%IF &SORT %THEN %DO;
  proc SORT;
    by descending Count;
  *;
%8: data: save various variable widths;
DATA _NULL_;
%local WLabel WCount WPERCENT;
retain WLabel WCount WPERCENT 0 ;
do until(EndoFile);
  set SERIES (keep = Label Count Percent) end = EndoFile;
  WLabel = max(WLabel ,length( Label ));
  WCount = max(WCount ,length(trim(left(put(Count ,8.0)))));
  WPERCENT = max(WPERCENT,length(trim(left(put(Percent ,5.1))))); end;
call symput('WLabel' , trim(left(put(WLabel ,2.))));
call symput('WCount' , trim(left(put(WCount ,8.))));
call symput('WPERCENT' , trim(left(put(WPERCENT,5.))));
stop; run;

%local LIB; %LET LIB = LIBRARY;
%IF &TESTING %THEN %DO;
%LET LIB = WORK;
%PUT WLabel = <&WLabel.> WCount=<&WCount.> WPERCENT=<&WPERCENT.> %END;

%7: data: save optimized dataset;
DATA &LIB..&SERIES. (label = "CHECKALL &SERIES." );
attrib
  %IF &BY_VAR ne %THEN %DO;
  Subset label = "subset: &BY_VAR = &BY_VALUE" %END;
%local LEN;
%LET LEN = %length(&TITLE.);
Title label = "&SERIES. Title"
  length = $ &LEN. format = Schar&LEN..
N_eq label =
  "N=&NNBRRESP data:&DATA Obs:&NNBROBS Resp:&PCNTRESP.%"
%local LENRESP;
%LET LENRESP = %val(2 + %length(&NNBRRESP.));
length = $ &LENRESP. format = Schar&LENRESP..
Chartype label = "&CHARTYPE"
  length = $ 8 format = Schar8.
Value label = "&SERIES. Variable"
  length = $ 8 format = Schar8.
Label label = "&LBL_LBL"
  length = $ &WLabel. format = Schar&WLabel..
Count label = "&LBLCOUNT." format = &WCount..0
Percent label = "&LBLPCENT." format = &WPERCENT..1;
;
retain Title "&TITLE."
  N_eq "N=&NNBRRESP."
  Chartype "&CHARTYPE";
do until(EndoFile);
  set SERIES
  %IF &BY_VAR ne %THEN %DO; (rename=(&BY_VAR = Subset)) %END;
  end = EndoFile;

Label = translate(Label,'|','');/*change <l> back to quote*/
***NbrResp = 100*Count/Percent;
output;
Title = '|.';
N_eq = '|.';
Chartype = '|.'; /*do until(EndoFile)*/end;
stop;

%8: proc: if wanted PRINT output dataset;
%IF &PRINT %THEN %DO; /*.....*/
TITLE3 "&SERIES.: &TITLE";
proc print data = &LIB..&SERIES. double label noobs;
%IF &BY_VAR ne %THEN %DO;
  by Subset ;
  id Subset ; %END;
proc CONTENTS data = &LIB..&SERIES.;
/*.....%IF &PRINT*/ %END;
%ENDONAC; run; /*.....*/ %MEND;

/* test data: enable by ending this line with slash '/' .....*/
options details mprint nocenter;
libname LIBRARY 'c:\saswinpd\sasuser\default';

%LET DATASET = SURVEY1; /*in-data: global macro-variable used by CHECKALL;

*Step 1: label the variables;

DATA LIBRARY.SURVEY1;
label Q01 = 'Fruit'
  Q02A = 'Apple' Q02B = 'Banana' Q02C = 'Cherry'
  N031 = 'One' N032 = 'Two' N033 = 'Three'
  X08C = 'Commercial' X08I = 'In House' X08M = 'Manufacturer';
input @ 1 Q01 Schar1.
  @ 2 Q02A Schar1. @ 3 Q02B Schar1. @ 4 Q02C Schar1.
  @ 2 N031 1. @ 3 N032 1. @ 4 N033 1.
  @ 5 X08C Schar1. @ 6 X08I Schar1. @ 7 X08M Schar1.; cards;

A100.X.
A110..X
A101X..
A001XXX
A100...
;
*Step 2: save CONTENTS of data set;

proc CONTENTS data = LIBRARY.SURVEY1 noprint
  out = LIBRARY.CONTENTS(keep = Name);

*Step 3: make data sets with series of multiple-response variables;

data LIBRARY.V002 LIBRARY.VM03 LIBRARY.VX08;
set LIBRARY.CONTENTS;
if substr(Name,1,3) = 'Q02' then output LIBRARY.V002;
if substr(Name,1,3) = 'N03' then output LIBRARY.VM03;
if substr(Name,1,3) = 'X08' then output LIBRARY.VX08;

* end SOP set-up .....*/
%CHECKALL(Q02);
%CHECKALL(N03,TRUE=1);
%CHECKALL(X08,TRUE='X');
%CHECKALL(Q02,BY_VAR=Q01);
/*..... END Test Data */

```