%SHOWCOMB: a macro to produce a data set with frequency of combinations of responses 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 because the number of possible response combinations is two to the power of the number of responses. This paper examines the SAS proc FREQ output data set from a cross-tabulation and discusses the issues in constructing a similar data set for multiple-response data with one variable containing the combination of responses. Issues related to labeling, storage and type of multiple-response variables are discussed.

The SHOWCOMB macro takes as parameters an output data set name which is the prefix of a series of variables containing the multiple-response data. The second parameter may be a list of the multiple-response variables, or the output data set provided by %CHECKALL. See Fehd (1996), (1997), %CHECKALL and %ARRAY.

INTRODUCTION

Simple questions may have complex answers when the question contains the phrase “Check all that apply”. This paper reviews the output data set of a proc FREQ cross-tabulation of a series of variables. This data set is used as a model to construct a macro which produces a standardized data set with the frequencies of the combinations of responses in multiple-response data.

The Answers: (Check all that apply)

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

Common values used for the meaning of ‘checked’ include: (Y,N), (T,F), etc. The example data uses numeric (1,0). A proc FREQ cross-tabulation is the easy first step in examining multiple-response data. Our proposed process requires saving the output data set.

```
Program 1

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

proc FREQ data = QUERIES;
 tables Q02A * Q02B * Q02C
   / list noprint out = FREQ;

proc PRINT data = FREQ label;

SAS output
OBS  Apple  Banana  Cherry  Count  Percent
---  -----  ------  ------  -----  -------
1    0      0       1       1       8.3333
2    0      1       0       2      16.6667
3    0      1       1       3      25.0000
4    1      0       0       3      25.0000
5    1      1       0       2      16.6667
6    1      1       1       1       8.3333
```

Our output is raw data: the values representing ‘checked’ and ‘not checked’ must be mentally translated while reading. Our next task is to replace the value for 'checked' in each variable with the variable label. This requires a new set of variables which are character with length of 40, the allowed length of labels. ‘Not checked’ is irrelevant and is changed to blank.

```
example intermediate output

Label  Label   Label
Q02A   Q02B    Q02C    Count    Percent
-----  ------  ------  -----    -------
               Cherry  1         8.3333
       Banana          2        16.6667
       Banana  Cherry  3        25.0000
Apple                  3        25.0000
Apple  Banana          2        16.6667
Apple  Banana  Cherry  1         8.3333
```

The last step is to concatenate the series of variables into one variable, compress, and delimit the labels with a comma. The data is sorted by descending Count.

```
example desired output

Combinations of Q02   Count    Percent
-------------------   -----    -------
Apple                 3        25.0000
Banana, Cherry        3        25.0000
Apple, Banana         2        16.6667
Banana                2        16.6667
Apple, Banana, Cherry 1         8.3333
Cherry                1         8.3333
```

Constraints of using cross-tabulation

When processing a cross-tabulation, SAS must allocate a matrix of N columns, where N is the number of variables. The number of matrix rows allocated is O(2**N) if data is binary-valued: (0,1) and O(3**N) if data contains missing (0,1,.). An additional consideration is the width of each column. The minimum space available for numeric variables is 2 or 3 bytes, depending on operating system. For multiple-response data with large numbers of variables, some optimization of both data storage and matrix size is necessary.

Numeric binary-valued data where the data is either zero, one or missing may be more effectively stored as character in one byte. The cost of this storage efficiency is that the data must be converted to numeric for usage in many SAS procedures.

Further efficiency may be realized by converting a series of binary-valued variables into an integer where each bit represents one variable. Again, the cost is conversion from compressed data to individual variables for analysis.

A major benefit of using an integer to store multiple-response data is that the width of the cross-tabulation matrix is reduced to one column. This reduces the chance that a production routine would fail.

Data compression

Each bit in an integer can be changed from zero to one on the condition that a contributing value is true. This routine has a parameter TRUE which can be changed to accept a value of numeric or character one. Each element in the array is tested for &TRUE and the bit changed accordingly. The exponent of 2 is the array dimension minus the Index; this changes bits from left to right, reflecting the left to right pass through the array. This allows the programmer to compare the variable values with the integer produced.

Macro code excerpt 1

```
%LET TRUE = 1;

%*3: data: prepare subset of DATA and create
     Number for FREQ which is the binary-value
     of all the variables with value = &TRUE;
DATA ZBINNMBR;
  set DATA;
  array CheckAll {*} &VAR_LIST;
  N = 0;
  do I = 1 to dim(CheckAll);
   if CheckAll{I} = &TRUE. then
                 N = sum(N, 2**(&DIM_VAR. - I));
                                  /*do I */ end;
```

Saving the labels

In this step the labels of the series are saved to an array of macro variables.

Macro code excerpt 2

```
%*3.2 create array of labels of series;
 length Label $ 40; drop Label;
 %DO I = 1 %TO &DIM_VAR;
  call label(&&VAR&I.,Label);
  call symput("LBL&I.",trim(left(Label)));
%END;
```

The data is now prepared for proc FREQ of a single variable: the integer containing the data from the series of variables. An output data is saved for the decompression step: ZFRQCOMB.

Decompression

Changing the integer to a character variable of combinations of labels is a two-step process. First the integer is changed to a character variable -- BinStrng -- containing zeros and ones. Then this binary string is used to concatenate the labels into one variable -- Label -- which contains the combinations.

In order to save data for large series, where more than 200 characters are needed for the combinations, a second process is carried out. An array of Labels is prepared. Lbls{I} always contains the same variable label or is blank. A shorter array of Columns is prepared -- Cols{} -- whose dimension is equal to the maximum number of items checked in the series. Each non-blank Lbls{} is moved to next empty Cols{}. Cols{1} will always contain one label, though it may be different in each combination.

Macro code excerpt 3

```
%*6. data: recode FREQ output:
     convert Number to Combinations;
DATA ZFRQCOMB;
 array Lbls {*}$ Lbl1-Lbl&DIM_LBL.;
 array Cols {*}$ Col1-Col&MAXCHKD.;
  set ZFRQCOMB;
  Delimitr = '';/*change to ',_' after 1st */
  %*6.1 change Number to binary string
  loop: change binary string to Label and Lbl*;
  BinStrng = put(N,binary&DIM_LBL..);
  %DO I = 1 %TO &DIM_LBL.;/*------------------*/
    if substr(BinStrng,&I,1) = '1'
     then do; Label = left(trim(Label)
                      !! Delimitr
                      !! "&&LBL&I.");
              Delimitr = ', ';
              Lbl&I    = "&&LBL&I." ;       end;
  /*................. *%DO I =1:&DIM_LBL*/ %END;
  if length(Label) = 200 then
             Label = '*' !! substr(Label,1,199);
  EmptyCol = 1;/*fill Cols from Lbls*/
  do I = 1 to dim(Lbls);/*--------------------*/
   if Lbls{I} ne ' ' then do;
                   Cols{EmptyCol}= Lbls{I};
                   EmptyCol = EmptyCol + 1; end;
/*...................... do I=1:dim(Lbls)*/ end;
```

Caution: a constraint on number of variables

Bytes contain eight bits. Eight bytes should allow this routine to handle 64 variables. SAS software (Win3.1 V6.11) stores very large integers (2**52+) in scientific notation (E-notation). This is not an accurate bit-string of the number representing the combinations. Therefore SHOWCOMB fails to return accurate results for more than 51 variables. A check of number of variables is provided.

CONCLUSION

When analyzing multiple-response data, a cross-tabulation from proc FREQ serves as a model to present the data. Users prefer to be able to read the labels of variables in each combination. Compressing the data saves system resources and allows large number of variables to be processed in a similar manner. When the data is decompressed, variable labels can be substituted for the variable values.

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), “%CHECKALL: a macro to produce a frequency of response data set from multiple-response data.” Proceedings of the Fourth Annual Conference of the SouthEast SAS Users Group, 393-398.

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

```
 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* MACRO: SHOWCOMB                                                       *
*                                                                       *
*  USAGE: 1) %SHOWCOMB(series-name);                                    *
*              *assumes CHECKALL data set exists for this series        *
*          2) %SHOWCOMB(series-name,LIST=var1 var2 var3 .. varN);       *
*          3) %SHOWCOMB(series-name,DATA=data set);                     *
*          4) %SHOWCOMB(series-name,TRIMCHAR=-);                        *
*          5) %SHOWCOMB(series-name,PRINT=1);                           *
*          6) %SHOWCOMB(series-name,BY_VAR=var-name);                   *
*          7) %SHOWCOMB(series-name,TRUE='Y');                          *
*                                                                       *
*  DESCRIPTION: Processing of Check-all-that-apply questions            *
*      output data set contains Combinations, Count Percent             *
*      and Columns containing each item of Combination                  *
*                                                                       *
*  PROCESS:                                                             *
*    1: macro: prepare ARRAY of variables                               *
*    2: macro: concatenate elements into VAR_LIST                       *
*    3: data: prepare subset of DATA and create Number for FREQ         *
*    4. proc: if BY_VAR present, then SORT data                         *
*    5. proc: FREQ of Number to output dataset                          *
*    6. data: recode FREQ output: convert Number to Combinations        *
*    7. data: save optimized dataset to library                         *
*    8. proc: if wanted CONTENTS and/or PRINT of saved data set         *
*    9. if LIST ne CHECKALL-DATA: process as in CHECKALL macro          *
*                                                                       *
*  NOTES: datasets created by %CHECKALL are named &SERIES.              *
*         these are input data sets to %ARRAY                           *
*                                                                       *
*  NOTES: max number of variables to analyze is 51                      *
*                                                                       *
*  KEYWORDS: APPEND array %ARRAY binary-coded call label()              *
*            data compression dim() dimension FREQ left()               *
*            multiple-response data put() trim()                        *
*                                                                       *
*  Author: Ron Fehd,                               RJF2@phpdLs1.em.cdc.gov *
*          Centers for Disease Control, and Prevention                  *
*          4770 Buford Hwy NE  MS-G25                     fax: 404/488-7667 *
*          Atlanta  GA  30341-3724                      voice: 404/488-4316 *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
```

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 SHOWCOMB(
 SERIES     /* name of series of variables, output data set prefix   **
            /* output data set name: &SERIES.CMB                      */
,LIST=DATA  /* list of variables                                      */
            /* default is DATA previously prepared                    */
            /* whose name is &SERIES: output from %CHECKALL macro     */
,LIBRARY=LIBRARY/* library name                                       */
,DATA=&DATASET./* DATASET is global variable, else hardcode here      */
            /* master DATA set name                                   */
            /*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!           */
            /* DO NOT USE &SERIES AS NAME OF THIS PARM                */
            /*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!           */
,LBL_LBL=Combinations of &SERIES./*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                                     */
,FORMAT=    /* format of single by-variable, use dot at end of format */
,CHARTYPE=QOOBARH/* Chart-Type in (barh barv tbl)                     */
,PRINT=0    /* WantPrint output? suppresses printing                  */
,TESTING=0  /* TESTING=1 enables explanatory printouts                */
,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'      */
);/*-------------------------------------------------------------------
RJF2:92Jul14 fixed to handle multiple by_vars
RJF2:92Jul17 fixed to handle multiple by_vars that are binary-valued
RJF2:96Apr09 when by_var present print ZFRQCOMB by Label &BY_VAR;
RJF2:96Jul00 polishing for SESUG
RJF2:96Oct17 max vars = 51 due to failure of sum at bit 52
.....................................................................*/
%*1: macro: prepare ARRAY of variables;
%IF "&LIST"="DATA" %THEN /*use CHECKALL dataset*/
      %ARRAY(VAR,DATA=&LIBRARY..&SERIES.,VAR=Value);
```

```
%ELSE %ARRAY(VAR,&LIST.);

%IF &DIM_VAR gt 51 %THEN %DO; %PUT !*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*;
                              %PUT Number of Vars exceeds 51: &DIM_VAR.;
                              %PUT SHOWCOMB aborted;
                              %PUT !*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*!*;
                              %GOTO ENDOMAC;                       %END;

%*2: macro: concatenate elements into VAR_LIST
            NOTE: LIST is tested below;
%MACRO  VAR_LIST;%DO I = 1 %TO &DIM_VAR; &&VAR&I. %END;           %MEND;
%local  VAR_LIST;
%LET    VAR_LIST = %VAR_LIST;

%IF &BY_VAR ne  %THEN %DO; TITLE4 "&SERIES. by &BY_VAR.";          %END;
%ELSE                 %DO; TITLE4 "&SERIES.";                      %END;

%*3: data: prepare subset of DATA and create Number for FREQ
     which is the binary-value of all the variables with value = &TRUE;
DATA ZBINNMBR;
 retain MaxChkd NmbrResp 0;
 do until(EndoFile);
  set &LIBRARY..&DATA.
   (keep = &BY_VAR. &VAR_LIST.)
   end   = EndoFile
   nobs  = NmbrObs;
  NmbrChkd = 0;
  array CheckAll {*} &VAR_LIST;
  N = 0;
  do I = 1 to dim(CheckAll);
   if CheckAll{I} = &TRUE. then do;    N = sum(N, 2**(&DIM_VAR. - I));
                                       NmbrChkd + 1;                end;
                                                          /*do I */ end;
  if NmbrChkd then do;                 output;
                                       MaxChkd = max(MaxChkd,NmbrChkd);
                                       NmbrResp + 1;/*if NmbrChkd*/ end;
                                             /*do until(EndoFile)*/ end;
%*3.2 create array of labels of series;
 %local DIM_LBL; %LET DIM_LBL = &DIM_VAR;
 length Label $ 40; drop Label;
 MwLabel = 0;/*save Maximum Width of Label*/
 %DO I = 1 %TO &DIM_VAR;
  call label(&&VAR&I.,Label);
  if index(Label,"&TRIMCHAR") then Label = left(substr(Label,
                                          index(Label,"&TRIMCHAR")+1));
  call symput("LBL&I.",trim(left(Label)));
  MwLabel = max(MwLabel,length(Label));                            %END;

%*3.3 create mac-vars of:;
 %local MWLABEL MAXCHKD NMBROBS NMBRRESP PCNTRESP;
 call symput("MWLABEL" ,trim(left(put(MwLabel ,2.))));
 call symput("MAXCHKD" ,trim(left(put(MaxChkd ,8.))));
 call symput("NMBROBS" ,trim(left(put(NmbrObs ,8.))));
 call symput("NMBRRESP",trim(left(put(NmbrResp,8.))));
stop; run;
%put MWLABEL  = <&MWLABEL.>;
%put MAXCHKD  = <&MAXCHKD.>;
%put NMBROBS  = <&NMBROBS.>;
%put NMBRRESP = <&NMBRRESP.>;
%LET PCNTRESP = %eval(100* &NMBRRESP /&NMBROBS);
%put PCNTRESP = <&PCNTRESP.>;

%*4. proc: if BY_VAR present, then SORT data;
%IF &BY_VAR. ne  %THEN %DO;
  proc SORT data = ZBINNMBR;
    by &BY_VAR.;                                                   %END;

%*5. proc: FREQ of Number to output dataset;
proc FREQ data = ZBINNMBR order = FREQ;
  tables N / out = ZFRQCOMB noprint;
  %IF  &BY_VAR. ne  %THEN %DO;
    by &BY_VAR.;                                                   %END;

%*6. data: recode FREQ output: convert Number to Combinations
      convert Number to binary string
      convert binary string to user-readable string in var Label & Lbl*
NOTE: long combinations may be truncated: max SAS char width is 200
      noted with asterisk in front of combinations;
DATA ZFRQCOMB;
 drop   Delimitr EmptyCol I
        Lbl1--Lbl&DIM_LBL.
        %IF not &TESTING %THEN N BinStrng;
        WLabel WCount WPercent
        WCol1--WCol&&MAXCHKD.;
 length Label    $ 200
        Delimitr $   2
        BinStrng $ &DIM_LBL.
  %DO I = 1 %TO &DIM_LBL.; Lbl&I.                                  %END;
  %DO I = 1 %TO &MAXCHKD.; Col&I.                                  %END;
                                  $ &MWLABEL.
  Col%eval(&MAXCHKD. + 1)         $ %length(&NMBRRESP.) /*:: Count  */
  Col%eval(&MAXCHKD. + 2)         $ 5                   /*:: Percent*/ ;
 array Lbls {*} $ Lbl1-Lbl&DIM_LBL.;
 array Cols {*} $ Col1-Col&MAXCHKD.;
 array WCol {*}   WCol1-WCol&MAXCHKD.;

 /* save to create mac-vars*/
 retain WLabel WCount WPercent
```

```
  %DO I = 1 %TO &MAXCHKD.;     WCol&I.                             %END;
  0 ;

 do until(EndoFile);/*------------------------------------------------*/
  set ZFRQCOMB end = EndoFile;

  /* initialize arrays */
  do I = 1 to dim(Lbls); Lbls{I} = ' ';                             end;
  do I = 1 to dim(Cols); Cols{I} = ' ';                             end;

  Label    = '';
  Delimitr = '';/*change to ',_' after first item*/

  %*6.1 change Number to binary string
  loop: change           binary string to Label and Lbl*;
  BinStrng = put(N,binary&DIM_LBL..);
  %DO I = 1 %TO &DIM_LBL.;/*------------------------------------------*/
    if substr(BinStrng,&I,1) = '1'
     then do;                         Label    = left(trim(Label)
                                                 !! Delimitr
                                                 !! "&&LBL&I.");
                                      Delimitr = ', ';
                                      Lbl&I    =    "&&LBL&I." ;    end;
  /*......................................... *%DO I =1:&DIM_LBL*/ %END;
  if length(Label) = 200 then         Label    = '*'
                                               !! substr(Label,1,199);
  EmptyCol = 1;/*fill Cols from Lbls*/
  do I = 1 to dim(Lbls);/*--------------------------------------------*/
   if Lbls{I} ne ' ' then do;         Cols{EmptyCol} = Lbls{I};
                                      WCol{EmptyCol}=max(WCol{EmptyCol},
                                                length(Cols{EmptyCol}));
    /*avoid error msg: */
    /* "array subscript out of bounds" */
    if EmptyCol lt &MAXCHKD. then     EmptyCol = EmptyCol + 1;      end;
  /*...........................................  do I=1:dim(Lbls)*/ end;

  %*copy numeric Count, Percent to character vars: Col*;
  Col%eval(&MAXCHKD. + 1) =      put(Count  ,%length(&NMBRRESP.).0) ;
  Col%eval(&MAXCHKD. + 2) = left(put(Percent,                  5.1));

  %*6.2 set-up for mac-vars: widths of variables, note all are retained;
  WLabel   = max(WLabel  ,length(Label));
  WCount   = max(WCount  ,length(left(trim(put(Count  ,8.0)))));
  WPercent = max(WPercent,length(left(trim(put(Percent,5.1)))));

  output; /*..................................*do until(EndoFile)*/ end;

 %*6.3 create mac-vars of widths;
 %local WLABEL WCOUNT WPERCENT;
 call symput('WLABEL'  ,trim(left(put(WLabel  ,3.))));
 call symput('WCOUNT'  ,trim(left(put(WCount  ,8.))));
 call symput('WPERCENT',trim(left(put(WPercent,8.))));
 %DO I = 1 %TO &MAXCHKD.;
  %local       WCOL&I.;
  call symput("WCOL&I.",trim(left(put(WCol&I.,2.))));              %END;
 stop;
run;

%local LIB; %LET LIB = &LIBRARY;

%IF &TESTING %THEN %DO;
 %LET LIB = WORK;
 %PUT WLABEL = <&WLABEL.> WCOUNT = <&WCOUNT.> WPERCENT = <&WPERCENT.>;
 %DO I = 1 %TO &MAXCHKD.;  %PUT WCOL&I. :: &&WCOL&I.;              %END;
                                                 /*%IF &TESTING */ %END;
%local SAVENAME; %LET SAVENAME = &SERIES.CMB;
%IF %length(&SAVENAME) gt 8 %THEN %LET SAVENAME=%substr(&SAVENAME,1,8);

%*7. data: save optimized dataset to library;
DATA &LIB..&SAVENAME. (label =
 %local LEN;
 %LET   LEN = %length(&TITLE.);
 %IF   &LEN. le 30  %THEN      "SHOWCOMB &TITLE.";
 %ELSE                         "SHOWCOMB %substr(&TITLE,1,30)";       );
 attrib
   %IF &BY_VAR ne  %THEN %DO;
   Subset     label  = "subset: &BY_VAR"                           %END;
   Title      label  = "&SERIES. Title"
              length = $ &LEN.     format = $char&LEN..
   N_eq       label  =
                  "N=&NMBRRESP data:&DATA Obs:&NMBROBS Resp:&PCNTRESP.%"
              %LET LENRESP  = %eval(2 + %length(&NMBRRESP.);
              length = $ &LENRESP. format = $char&LENRESP..
   Chartype   label  = "&CHARTYPE"
              length = $ 8         format = $char8.
   Label      label  = "&LBL_LBL"
              length = $ &WLABEL.  format = $char&WLABEL..
   Count      label  = "&LBLCOUNT." format = &WCOUNT..0
   Percent    label  = "&LBLPCENT." format = &WPERCENT..1
   %DO I = 1 %TO &MAXCHKD.;
   Col&I.     label  = "Col &I."
              length = $ &&WCOL&I. format = $char&&WCOL&I...       %END;
              /*PC: three dots in $charN. mainframe ???   ... */
   Col%eval(&MAXCHKD. + 1)
              label  = "Count Col %eval(&MAXCHKD. + 1)"
              length = $ %length(&NMBRRESP.)
                                   format = $char%length(&NMBRRESP.).
   Col%eval(&MAXCHKD. + 2)
```

```
              label  = "Percent Col %eval(&MAXCHKD. + 2)"
              length = $ &WPERCENT. format = $char&WPERCENT..
   ;
 retain Title    "&TITLE."
        N_eq     "N=&NMBRRESP."
        Chartype "&CHARTYPE";
 do until(EndoFile);/*------------------------------------------------*/
   set ZFRQCOMB
   %IF &BY_VAR ne  %THEN %DO;
   (rename=(&BY_VAR = Subset))                                     %END;
    end = EndoFile;
   Label    = translate(Label,'!',"'");/*change <!> back to squote*/
***NmbrResp   = 100*Count/Percent;
   output;
   Title    = '.';
   N_eq     = '.';
   Chartype = '.';/*......................... *do until(EndoFile)*/ end;
stop;

%*8. proc: if wanted CONTENTS and/or PRINT of saved data set;
%IF &TESTING %THEN %DO; proc CONTENTS data = &SYSLAST.;            %END;

%IF &PRINT %THEN %DO;/*-----------------------------------------------*/
proc PRINT data = &LIB..&SAVENAME double noobs label;
  var
   %IF &TESTING %THEN       N BinStrng                                ;
   %IF &WLABEL gt 100 %THEN Col:                                      ;
   %ELSE                    Label Count Percent                     ; ;
  %IF  &BY_VAR. ne  %THEN %DO;
    by Subset;*&BY_VAR.;
    id Subset;*&BY_VAR.;
    %IF &FORMAT. ne %THEN %DO;
      format &BY_VAR. &FORMAT.;                                    %END;
                                        /**%IF BY-VAR not missing*/ %END;
 /*..................................... *%IF PRINT ............*/ %END;

%*9. if LIST ne CHECKALL-DATA: process as in CHECKALL macro;
%IF "&LIST." = "DATA"
 and &BY_VAR =  %THEN %DO; %PUT EXIT: LIST = CHECKALL DATA;        %END;
%ELSE %DO;*routine similar to CHECKALL;                            %END;
%ENDOMAC:  run;/*....................................SHOWCOMB..*/ %MEND;
 /*- SHOWCOMB test data: enable by ending line with slash ‘/’  ------**
options details mprint nocenter;
libname LIBRARY 'c:\saswinpd\sasuser';*default*;
%LET DATASET = SURVEY2;
*Step 1: label the variables;
DATA LIBRARY.SURVEY2;
label
Q04A     = "Q04a: Primary Classification"
Q04BCOM  = "Q04B: Blood Bank - Community"
Q04BREG  = "Q04B: Blood Bank - Regional"
Q04BPLA  = "Q04B: Blood Bank - Blood/Plasma center"
Q04BARC  = "Q04B: Blood Bank - American Red Cross"
Q04BPRI  = "Q04B: Blood Bank - Privately owned"
Q04BMIL  = "Q04B: Blood Bank - Military (Federal)"
Q04BHOS  = "Q04B: Blood Bank - Hospital blood bank"
Q04BOTR  = "Q04B: Blood Bank - Other"
Q04CCIT  = "Q04C: Hospital - City"
Q04CCNT  = "Q04C: Hospital - County"
Q04CSTA  = "Q04C: Hospital - State"
Q04CDIS  = "Q04C: Hospital - District"
Q04CCOM  = "Q04C: Hospital - Community"
Q04CREG  = "Q04C: Hospital - Regional"
Q04CMIL  = "Q04C: Hospital - Military (Federal)"
```

```
Q04CVET  = "Q04C: Hospital - Veterans Administration"
Q04CPRI  = "Q04C: Hospital - Privately owned"
Q04CUNI  = "Q04C: Hospital - University"
Q04CHMO  = "Q04C: Hospital - HMO owned & operated"
Q04CREL  = "Q04C: Hospital - Religious associated"
Q04COTR  = "Q04C: Hospital - Other"
; input
@ 1 Q04A    $char1. @ 3 Q04BCOM $char1. @ 4 Q04BREG $char1.
@ 5 Q04BPLA $char1. @ 6 Q04BARC $char1. @ 7 Q04BPRI $char1.
@ 8 Q04BMIL $char1. @ 9 Q04BHOS $char1. @10 Q04BOTR $char1.
@12 Q04CCIT $char1. @13 Q04CCNT $char1. @14 Q04CSTA $char1.
@15 Q04CDIS $char1. @16 Q04CCOM $char1. @17 Q04CREG $char1.
@18 Q04CMIL $char1. @19 Q04CVET $char1. @20 Q04CPRI $char1.
@21 Q04CUNI $char1. @22 Q04CHMO $char1. @23 Q04CREL $char1.
@24 Q04COTR $char1.
;cards;
B 10000010 0000000000000
H 00000000 0000110000000
H 00000000 1100000000000
H 00000000 0000110000000
B 11000000 0000000000000
H 00000000 0000000100000
B 11000000 0000000000000
H 00000000 0000110000000
B 10000000 0000000000000
B 10000010 0000000000000
H 00000000 0000000100000
H 00000000 0000000000000
H 00000000 0000000000000
B 10000010 0000000000000
H 00000000 0000000100000
H 00000000 1100000000000
H 00000000 0000110000000
H 00000000 0000000100000
B 10000010 0000000000000
; *Step 2: save CONTENTS of data set;
proc CONTENTS data = LIBRARY.&DATASET. noprint
              out  = LIBRARY.CONTENTS(keep = Name);
*Step 3: create and save data sets with series of variables;
data LIBRARY.VQ04  LIBRARY.VQ04B  LIBRARY.VQ04C;
 set LIBRARY.CONTENTS;
 if substr(Name,1,3) = 'Q04'  then output LIBRARY.VQ04;
 if substr(Name,1,4) = 'Q04B' then output LIBRARY.VQ04B;
 if substr(Name,1,4) = 'Q04C' then output LIBRARY.VQ04C;
 * end CHECK-ALL set-up **********************************************;
*Step 4: run %CHECKALL:
         output dataset contains only variables used in the series,
         used as input for SHOWCOMB;
%CHECKALL(Q04);
%CHECKALL(Q04B,TRIMCHAR=-);
%CHECKALL(Q04C,TRIMCHAR=-);
 * end SHOWCOMB SOP set-up *********************************************;
%SHOWCOMB(Q04);
%SHOWCOMB(Q04,BY_VAR=Q04A,TRIMCHAR=-);
%SHOWCOMB(Q04B,TRIMCHAR=-);
%SHOWCOMB(Q04C,TRIMCHAR=-);
*show intermediate vars*:;
*SHOWCOMB(Q04B,BY_VAR=Q04A,TRIMCHAR=-,TESTING=1);
*user-provided list:;
*SHOWCOMB(Q04B,LIST = Q04BCOM Q04BPLA Q04BPRI Q04BREG, TRIMCHAR=-);
run;/*................................................ end test data */
```