PROC DATACHK Revisited: The DATACHK Macro

Jeffrey M. Abolafia
University of North Carolina

Introduction

PROC DATACHK, a SUGI Supplemental Library procedure in Version 5 of the SAS® System, provided a useful method for quickly checking numeric data in a selected SAS data set. For each numeric variable, PROC DATACHK listed the number of nonmissing observations, the number of missing observations, the five lowest values, the five highest values, and the number of observations having six prespecified missing values. While this information is available in PROC UNIVARIATE, a major advantage of PROC DATACHK was that output was presented concisely: one line per variable. With the release of Version 6 of the SAS System, the SUGI Supplemental Library procedures were discontinued. Since that time, a new procedure to replace PROC DATACHK has not been developed.

The DATACHK macro attempts to simulate PROC DATACHK. Like PROC DATACHK, the DATACHK macro was developed to quickly check numeric variables in a selected SAS data set. For each numeric variable, DATACHK concisely lists the number of nonmissing observations, the number of missing observations, the five lowest values, and the five highest values (the DATACHK macro does not list the six prespecified missing values).

Usage

The macro DATACHK contains two keyword parameters:

DATA	The name of an existing SAS data set. This can be a permanent or work data set. You can specify a one or two level name. This parameter is required.
VAR	A list of numeric variables separated by spaces. If the VAR parameters not specified then all numeric variables are processed. This parameter is optional.

Below are two example of using the macro DATACHK. Example 1 uses the macro without the optional VAR parameter. Example two invokes the macro using the VAR parameter.

```
%DATACHK(DATA=IN.FINAL)

%DATACHK(DATA=IN.FINAL,VAR=LDL HDL AGE)
```

Printed Output

The DATACHK macro prints the following information for each numeric variable in a selected SAS data set:

1. the number of nonmissing observations
2. the number of missing observations
3. the five largest values
4. the five smallest values
5. the variable label (if one exists).

Below is sample output from the DATACHK macro.

```
VARIABLE        N    MISSING                  5 Smallest Values                                      5 Largest Values

PACKNO        698        0  |    2010      2014      3015      3041      3049 |  95019     95022     95023     95039     96012
PACKET NUMBER               |                                                 |
                            |                                                 |
EVDATE         72      626  | 01JUL76   26JUL76   31JUL76   01AUG76   02AUG76 | 11JUN77   15JUN77   17JUN77   18JUN77   23JUN77
                            |                                                 |
                            |                                                 |
DRINKER       694        4  |       0         1                               |                                     0         1
                            |                                                 |
_CAL          698        0  |438.4149  565.9099  650.3849  802.2349  867.2399 | 4125.64   4141.15   4176.87  4281.039  4916.199
                            |                                                 |
QLDL          698        0  |       0         1         2         3           |                     0         1         2         3
RANK FOR VARIABLE LDL       |                                                 |
```

Additional Features

1. DATACHK performs some error checking. If the input data set does not exist or if variables specified on the VAR parameter do not exist, DATACHK prints an appropriate message and will stop executing.

2. If a variable has a date format of DATE7., DDMMYYw., MMDDYYW., or YYMMDDw., then the five largest and smallest values are printed with the DATE7. format.

3. If a variable has a time format of TIMEw. then the five largest and smallest values are printed with the TIME8. format.

4. Variables that do not have a date or time format are displayed with the BEST10. format.

5. Variable labels are printed if they exist.

REFERENCES

SAS Institute Inc. (1983), SUGI Supplemental Library User's Guide 1983 Edition. Cary, NC: SAS Institute Inc.

Acknowledgments

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. Other brand and product names are registered trademarks of their respective companies.

Jeffrey Abolafia
University of North Carolina
Department of Biostatistics
CB #8030, 203 Nations Bank Plaza
Chapel Hill, NC 27514
uccjma.cscc@mhs.unc.edu

APPENDIX

```
%MACRO DATACHK(DATA=,VAR=);

 %*****************************************************************;
 %*** NAME: DATACHK                                            ***;
 %***                                                          ***;
 %*** VERSION: 1                                               ***;
 %***                                                          ***;
 %*** FUNCTION:                                                ***;
 %***                                                          ***;
 %*** DATACHK can be used to quickly check numeric variables   ***;
 %*** in a selected SAS data set. It is similar to PROC DATACHK***;
 %*** in SAS Version 5.18 . For each numeric variable, DATACHK ***;
 %*** lists the number of nonmissing observations, the number  ***;
 %*** of missing observations, the five lowest values, and the ***;
 %*** five highest values. Variable labels are also included.  ***;
 %*** DATACHK runs under SAS Version 6.11 or later. A copy of  ***;
 %*** DATACHK that runs under earlier version of SAS can be    ***;
 %*** obtained from the author.                                ***;
 %***                                                          ***;
 %***   PARAMETERS:                                            ***;
 %***                                                          ***;
 %***     DATA     The name of an existing SAS data set.       ***;
 %***              This can be a permanent or work data        ***;
 %***              set. You can specify a one or two level     ***;
 %***              name. This parameter is required.           ***;
 %***                                                          ***;
 %***     VAR      a list of numeric variables, seperated      ***;
 %***              by spaces, to be checked. If VAR is not     ***;
 %***              specified then all numeric variables are    ***;
 %***              processed.                                  ***;
 %***                                                          ***;
 %***   EXAMPLES:                                              ***;
 %***                                                          ***;
 %***               %DATACHK(DATA=IN.FINAL)                    ***;
 %***               %DATACHK(DATA=IN.FINAL,VAR=LDL HDL AGE)    ***;
 %***                                                          ***;
 %***   AUTHOR: Jeffrey Abolafia                               ***;
 %***                                                          ***;
 %***   NOTE:   Uses macros VARLIST and COUNT                  ***;
 %***                                                          ***;
 %*****************************************************************;
 options nosource nonotes ;

   %put ** MACRO DATACHK VERSION 1 IS EXECUTING ** ;

   %let var=%upcase(&var) ;

  /* check that DATA exists */
   %if &data= %then %let ex=NO ;
   %else %do ;
      %if %sysfunc(exist(&data)) %then %let ex=YES;
      %else %let ex=NO ;
   %end ;
   %if &ex=NO %then %do ;
     %put THE INPUT DATA SET DOES NOT EXIST DATCHK WILL STOP EXECUTING;
     %goto nodata ; ** this will end execution ** ;
   %end;

   /* store all numeric variables in a global macro variable */

   %varlist(data=&data,vars=numvars,type=n)

   /* if no numeric variables are selected then end execution */

   %if &numvars=  %then %do ;
      %PUT THERE ARE NO NUMERIC VARIABLES IN &DATA DATACHK WILL STOP EXECUTION;
      %goto nodata ;  ** end execution ** ;
   %end ;

   /* count number of numeric variables and store in macro variable */

%COUNT(LIST=&numvars,NUM=nvars)

  /* If VAR= parameter is specified check for the existance
      of variables specified
  */

  %if &var ne %then %do;
     %let c=1;
     %let prob=0 ;
     %do %until(%scan(&var,&c)= ) ;
         %let v = %scan(&var,&c) ;
         %if %index(&numvars,&v)=0 %then %do;
             %let prob=1 ;
             %put THE VARIABLE &V NOT FOUND IN &DATA DATACHK WILL STOP EXECUTION;
         %end;
         %let c=%eval(&c + 1) ;
      %end ;
      %if &prob=1 %then %goto nodata ;  ** end execution ** ;
   %end ;

   /* end of variable check algorithm */

   /* calculate N, # missing, 5 lowest and highest for each
      numeric variable
   */

 data outx(keep=name n _m h1-h5 l1-l5);
   set &data(keep= &numvars)  end=eof ;

   /* store Ns, # missing, 5 highest and lowest in temporary arrays */
   /* Note: elements in temporary arrays are automatically retained */
  array vars{&nvars} &numvars ;
  array ns{&nvars} _TEMPORARY_ ;
  array m{&nvars} _TEMPORARY_  ;
  array la{&nvars} _TEMPORARY_ ;
  array lb{&nvars} _TEMPORARY_ ;
  array lc{&nvars} _TEMPORARY_ ;
  array ld{&nvars} _TEMPORARY_ ;
  array le{&nvars} _TEMPORARY_ ;
  array ha{&nvars} _TEMPORARY_ ;
  array hb{&nvars} _TEMPORARY_ ;
  array hc{&nvars} _TEMPORARY_ ;
  array hd{&nvars} _TEMPORARY_ ;
  array he{&nvars} _TEMPORARY_ ;

do i = 1 to &nvars ;  /* begin loop for each variable */

  ns{i} + (vars{i} > .z);      /* no. non-missing */
  m{i} + (vars{i} <=.z) ;      /* no. missing     */

  if vars{i} > .z then do; /* find 5 lowest & highest if not missing */
    if vars{i} > he{i} then do;
       ha{i} = hb{i} ;
       hb{i} = hc{i} ;
       hc{i} = hd{i} ;
       hd{i} = he{i} ;
       he{i} = vars{i} ;
    end  ;
    else if hd{i} < vars{i} < he{i} then do;
       ha{i} = hb{i} ;
       hb{i} = hc{i} ;
       hc{i} = hd{i} ;
       hd{i} = vars{i} ;
    end  ;
    else if hc{i} < vars{i} < hd{i} then do;
       ha{i} = hb{i} ;
       hb{i} = hc{i} ;
       hc{i} = vars{i} ;
    end  ;
    else if hb{i} < vars{i} < hc{i} then do;
       ha{i} = hb{i} ;
       hb{i} = vars{i} ;
    end  ;
    else if ha{i} < vars{i} < hb{i} then do;
       ha{i} = vars{i} ;
    end  ;

    if .Z < vars{i} < la{i} then do;
       le{i} = ld{i} ;
       ld{i} = lc{i} ;
       lc{i} = lb{i} ;
       lb{i} = la{i} ;
       la{i} = vars{i} ;
     end;
    else if (vars{i} > .z ) & (la{i} <= .z) then la{i} = vars{i} ;
    else if (la{i} < vars{i} < lb{i}) & (lb{i} >.z) then do;
       le{i} = ld{i} ;
       ld{i} = lc{i} ;
       lc{i} = lb{i} ;
       lb{i} = vars{i} ;
    end;
    else if (vars{i} > la{i} ) & (la{i} > .z) & (lb{i} <= .z) then
        lb{i} = vars{i} ;
    else if (lb{i} < vars{i} < lc{i}) & (lc{i} >.z) then do;
       le{i} = ld{i} ;
       ld{i} = lc{i} ;
       lc{i} = vars{i} ;
    end;
    else if (vars{i} > lb{i} ) & (lb{i} >.z ) & (lc{i} <= .z) then
        lc{i} = vars{i} ;
    else if (lc{i} < vars{i} < ld{i}) & (ld{i} >.z ) then do;
       le{i} = ld{i} ;
       ld{i} = vars{i} ;
    end;
    else if (vars{i} > lc{i} ) & (lc{i} >.z) & (ld{i} <= .z) then
       ld{i} = vars{i} ;
    else if (ld{i} < vars{i} < le{i}) & (le{i} >.z ) then
      le{i} = vars{i} ;
    else if (vars{i} > ld{i} ) & (ld{i} > .z) & (le{i} <= .z) then
       le{i} = vars{i} ;

  end;
end;

 /* restructure data to 1 record per variable */

  length name $8 ;
  if (eof) then do i=1 to &nvars ;
     call vname(vars{i},name) ;
     n = ns{i} ;
     _m = m{i} ;
     l1 =la{i} ;
     l2 =lb{i} ;
     l3 =lc{i} ;
     l4 =ld{i} ;
     l5 =le{i} ;
     h1 =ha{i} ;
     h2 =hb{i} ;
     h3 =hc{i} ;
     h4 =hd{i} ;
     h5 =he{i} ;
     output ;
  end ;
run;

   /* Merge in dictionary table containing variable labels and
      formats(SASHELP.VCOLUMN). Then reformat variables for
      display purposes.
   */

data _null_ ;
   merge outx
         sashelp.vcolumn(keep=libname memname type name label format
            where=(libname="&libref" & memname="&dsn" & type='num'
                   %if &var ne  %then & indexw("&var",name) > 0  ;
                   )) ;

   /* convert 5 highest and lowest to appropriate character format */

    array ovar{10} l1-l5 h1-h5 ;
    array nvar{10} $10 a1-a5 b1-b5 ;

    do i= 1 to 10;

      if ovar{i}<=.z then nvar{i}=' ' ;   ** missing ** ;

      /* if original variable had a date format of DATE7.
         DDMMYYw. MMDDYYw. or YYMMDDw. then format with the
         DATE7. format
      */

      else if (substr(format,1,4) ='DATE') or
              (substr(format,1,6) in('DDMMYY','MMDDYY','YYMMDD'))
        then nvar{i}='   ' || put(ovar{i},date7.) ;

      /* if original variable had a time format of TIMEw.
         then format with the TIME8. format
      */

      else if (substr(format,1,4) ='TIME')
        then nvar{i}='  ' || put(ovar{i},time8.)  ;

      /* otherwise format with the best10. format */

      else nvar{i}=right(put(ovar{i},best10.)) ;

    end ;

    /* send output to standard print file */
   file print notitles ls=173 PS=60 header=h ;

       put @1 name $8. @11  n 7. @22 _m 7.  @42 '|' @44
           (a1-a5) ($10., +2) +1 '|' +1 (b1-b5) ($10., +2)  / @1 label @42 '|'
           @103 '|' / @42 '|' @103 '|' ;

       return;
       h:
       put @58 "NUMERIC DATA CHECKS FOR DATA SET %upcase(&DATA)" ///
          @1 'VARIABLE' @16 'N' @22 'MISSING'
             @66 '5 SMALLEST VALUES'
             @124 '5 LARGEST VALUES' / ;
       return ;
     run;

     ** delete data set created during macro execution ** ;
     proc printto log=work.temp.temp.log new ;
     run;
     proc datasets library=work ;
        delete outx;
     run;
     quit ;
     proc printto  ;
     run;

     %nodata:
     options source notes ;
%MEND DATACHK ;   /*  END OF MACRO DATACHK */


   /* auxillary macros */

%macro varlist(data=,vars=,type=) ;

 %*****************************************************************;
 %*** VARLIST CREATE A GLOBAL MACRO VARIABLE THAT HOLDS A LIST ***;
 %*** OF ALL VARIABLES IN A SELECTED DATA SET WHERE:           ***;
 %***                                                          ***;
 %***   DATA       THE NAME OF AN EXISTING SAS DATA SET.       ***;
 %***              THIS CAN BE A PERMANENT OR WORK DATA        ***;
 %***              SET. YOU CAN SPECIFY A ONE OR TWO LEVEL     ***;
 %***              NAME                                        ***;
 %***                                                          ***;
 %***   VARS =     THE NAME OF THE GLOBAL MACRO VARIABLE WHICH ***;
 %***              WILL HOLD THE LIST OF VARIABLES IN THE      ***;
 %***              DATASET LIBREF.DSN                          ***;
 %***                                                          ***;
 %***   TYPE =     IF YOU WOULD LIKE ONLY A LIST OF NUMERIC    ***;
 %***              VARIABLES SET TYPE EQUAL TO N . IF YOU      ***;
 %***              WOULD LIKE A ONLY A LIST OF CHARACTER       ***;
 %***              VARIABLES SET TYPE EQUAL TO C. THE DEFAULT  ***;
 %***              IS A LIST OF ALL VARIABLES.                 ***;
 %***                                                          ***;
 %***                                                          ***;
 %***  EXAMPLE:                                                ***;
 %***                                                          ***;
 %***               LIBNAME SC V604 'C:\BIOS111\SASDATA' ;     ***;
 %***               %VARLIST(DATA=SC.CLASS,VARS=NVARS,TYPE=N)  ***;
 %*****************************************************************;

   %let data=%upcase(&data) ;

  %global libref dsn ;
  %if %index(&data,.)=0 %then %do ; /* one level */
     %let libref= WORK ;
     %let dsn = &data ;
  %end ;
 %else %do ;
    %let libref=%scan(&data,1,'.') ;
    %let dsn=%scan(&data,2,'.') ;
 %end ;


  ** Create a global macro variable that will hold     **
  ** the list of variables from the data set specified **
  ** by the libref and dsn parameters. The name of the **
  ** macro variable is value specified for the VARS    **
  ** macro variable                                    ** ;

%global &vars ;
proc sql noprint ;
  select name
  into :&vars seperated by ' '
  from sashelp.vcolumn
  where libname="&libref" and memname="&dsn"
        %if %upcase(&type)=N %then & type='num' ;
         %else %if %upcase(&type)=C %then & type='char' ;
         %if &var ne  %then  & indexw("&var",name) > 0 ;
        ;
quit ;
%put &nvars ;

%mend varlist ;


 %MACRO COUNT(LIST=,NUM=);

 %*****************************************************************;
 %*** COUNTS NUMBER OF ELEMENTS IN A LIST AND CREATES A MACRO  ***;
 %*** VARIABLE TO HOLD THE VALUE                               ***;
 %***                                                          ***;
 %***   LIST =     A MACRO VARIABLE CONTAINING A LIST          ***;
 %***              OF ELEMENTS                                 ***;
 %***   NUM  =     NEW GLOBAL MACRO VARIABLE WHICH             ***;
 %***              HOLDS THE NUMBER OF ELEMENTS IN LIST        ***;
 %***                                                          ***;
 %***  EXAMPLE:     %COUNT(LIST=&COVAR,NUM=NCOVAR)             ***;
 %*****************************************************************;


 %*** CREATE GLOBAL VARIABLE TO PASS BACK TO CALLING MACRO ***;
 %GLOBAL &NUM;

 %*** SET TO FIRST ELEMENT OF LIST AND ENTER LOOP IF NOT EMPTY ***;
 %LET &NUM=0;
 %LET WORD=%SCAN(&LIST,&&&NUM+1);

 %DO %WHILE(&WORD NE);
    %LET &NUM=%EVAL(&&&NUM+1);
    %LET WORD=%SCAN(&LIST,&&&NUM+1);
    %END;

 %MEND COUNT;

 /* END OF AUXILLARY MACROS */
```