

Paper 048-2010

Dropping Automatically Variables with Only Missing Values

Selvaratnam Sridharma, Census Bureau, Washington, D.C.

ABSTRACT

Sometimes all the values of some variables in a SAS[®] dataset are missing or null, and we would like to drop these variables to save disk space. Here we introduce a macro %DROPMISS that can automatically identify and drop SAS variables that have only missing or null values. You have the option of not dropping the variables you do not want to drop by using a parameter in the %DROPMISS macro.

INTRODUCTION

When all the values in some numeric or character variables in a SAS dataset are missing, dropping these variables can save a large amount of disk space. Using the macro %DROPMISS developed here, one can drop automatically the variables that are always missing. Sometimes, you want to keep some variables even though they are always missing. Using %DROPMISS macro, you will be able to do that as explained below. When you run this macro it will write in the log the names of the variables if they are dropped. If none of the variables are dropped it will write so in the log.

%DROPMISS MACRO

The code for %DROPMISS macro is given in the Appendix. Here we will show how to use the %DROPMISS macro using the following dataset DSIN.

```
data DSIN;
  input a s d z $ c $ x $ y ;
datalines;
1 . 3 ab pq . .
. . . . xy . .
3 . 3 ln . . .
;
```

In the dataset DSIN, all the values of the numeric variable s and y are null, and all the values of the character variable x are missing. If we want to drop all the variables that are always missing or null, we will use the %DROPMISS macro as follows.

```
%DROPMISS(DSIN, DSOUT)
```

When you run this macro, output dataset DSOUT will be produced. The dataset DSOUT is the same as DSIN except that it will not have the variables s, y, and x. These variables are always missing or always null in the dataset DSIN.

But, if we want to keep the variables y and x even though they are always missing or null, then you use the %DROPMISS as follows.

```
%DROPMISS(DSIN,DSOUT, nodrop= y x)
```

Only the variable s will be dropped.

If you do not want to drop any of the numeric variables, you can use

```
%DROPMISS (DSIN,DSOUT, nodrop= _NUMERIC_ ).
```

If you do not want to drop any of the character variables, you can use

```
%DROPMISS (DSIN,DSOUT, nodrop= _CHARACTER_ ).
```

A program in Reference 1 named '**Delete variables that have only missing values**' can be used to Identify and remove any variables, character or numeric, which has only missing values. But %DROPMIS is more efficient than the program in Reference 1. The following tables shows that it takes %DROPMISS macro less time than the program in Reference 1.

	Size (Bytes)	Time
Program in Sample 53	567889	13.2
%DROPMISS	567889	10.7

When you use %DROPMISS macro, you have the option of not dropping the variables you do not want to drop and, the names of those variables that are dropped will be written to the log, but the program in Reference 1 will not do these.

CONCLUSION

%DROPMISS macro is very efficient, and could save you a huge amount of space if some of the variables in a SAS dataset are always missing or null and they are dropped.

REFERENCES

1. Sample 24612: Delete variables that have only missing values.

12 June. 2009 <http://support.sas.com/ctx/samples/index.jsp?sid=53&tab=code>

ACKNOWLEDGMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Selvaratnam Sridharma
 Economic Planning and Coordination Division
 U.S. Bureau of the Census
 Address
 Washington, DC 20233-6100
 301-763-6774
 Email: selvaratnam.sridharma@census.gov

APPENDIX: %DROPMISS

```

/*****/
options nomprint noSYMBOLGEN MLOGIC;
/*****/

%macro DROPMISS( DSNIN      /* name of input SAS dataset
*/
                    , DSNOUT /* name of output SAS dataset
*/
                    , NODROP= /* [optional] variables to be omitted from dropping even if
they have only missing values */
                    ) ;

    /* PURPOSE: To find both Character and Numeric the variables that have only
missing values and drop them if
    *           they are not in &NODROP
    *
    * NOTE:     if there are no variables in the dataset, produce no variables
processing code
    *
    *
    * EXAMPLE OF USE:
    *           %DROPMISS( DSNIN, DSNOUT )
    *           %DROPMISS( DSNIN, DSNOUT, NODROP=A B C D--H X1-X100 )
    *           %DROPMISS( DSNIN, DSNOUT, NODROP=_numeric_ )
    *           %DROPMISS( DSNIN, DSNOUT, NODROP=_character_ )
    */

%local I ;

%if "&DSNIN" = "&DSNOUT"
%then %do ;
    %put /-----\ ;
    %put | ERROR from DROPMISS: | ;
    %put | Input Dataset has same name as Output Dataset. | ;
    %put | Execution terminating forthwith. | ;
    %put \-----/ ;

    %goto L9999 ;
%end ;

/*****/
/* begin executable code
/*****/

/*=====*/
/* Create dataset of variable names that have only missing values
/* exclude from the computation all names in &NODROP
/*=====*/

proc contents data=&DSNIN( drop=&NODROP ) memtype=data noprint out=_cntnts_( keep=
name type ) ; run ;

%let N_CHAR = 0 ;
%let N_NUM = 0 ;

data _null_ ;
    set _cntnts_ end=lastobs nobs=nobs ;

```

```

if nobs = 0 then stop ;

n_char + ( type = 2 ) ;
n_num  + ( type = 1 ) ;

/* create macro vars containing final # of char, numeric variables */

if lastobs
then do ;
    call symput( 'N_CHAR', left( put( n_char, 5. ))) ;
    call symput( 'N_NUM' , left( put( n_num , 5. ))) ;
end ;
run ;

/*=====*/
/* if there are no variables in dataset, stop further processing
/*=====*/

%if %eval( &N_NUM + &N_CHAR ) = 0
%then %do ;
    %put /-----\ ;
    %put | ERROR from DROPMISS: | ;
    %put | No variables in dataset. | ;
    %put | Execution terminating forthwith. | ;
    %put \-----/ ;

    %goto L9999 ;
%end ;

/*=====*/
/* put global macro names into global symbol table for later retrieval
/*=====*/
%LET NUM0 =0;
    %LET CHAR0 = 0;
    %IF &N_NUM >0 %THEN %DO;
%do I = 1 %to &N_NUM ;
    %global NUM&I ;
%end ;
%END;
    %if &N_CHAR > 0 %THEN %DO;
%do I = 1 %to &N_CHAR ;
    %global CHAR&I ;
%end ;
%END;
/*=====*/
/* create macro vars containing variable names
/* efficiency note: could compute n_char, n_num here, but must declare macro names
to be
global b4 stuffing them
/*
/*=====*/

proc sql noprint ;
    %if &N_CHAR > 0 %then %str( select name into :CHAR1 - :CHAR&N_CHAR from
_cntnts_ where type = 2 ; ) ;

    %if &N_NUM > 0 %then %str( select name into :NUM1 - :NUM&N_NUM from
_cntnts_ where type = 1 ; ) ;
quit ;

/*=====*/
/* Determine the variables that are missing

```

```

/*
/*=====*/

%IF &N_CHAR > 1 %THEN %DO;
  %let N_CHAR_1 = %EVAL(&N_CHAR - 1);
%END;

Proc sql ;
  select %do I= 1 %to &N_NUM; max (&&NUM&I) , %end; %IF &N_CHAR > 1 %THEN %DO;
         %do I= 1 %to &N_CHAR_1; max(&&CHAR&I), %END; %end; MAX(&&CHAR&N_CHAR)
into
         %do I= 1 %to &N_NUM; :NUMMAX&I , %END; %IF &N_CHAR > 1 %THEN %DO;
         %do I= 1 %to &N_CHAR_1; :CHARMAX&I,%END; %END; :CHARMAX&N_CHAR
from &DSNIN;

quit;

/*=====*/
/* initialize DROP_NUM, DROP_CHAR global macro vars
/*=====*/

%let DROP_NUM      = ;
%let DROP_CHAR     = ;

%if &N_NUM > 0 %THEN %DO;
DATA _NULL_;

  %do I = 1 %to &N_NUM ;
    %IF &&NUMMAX&I =. %THEN %DO;
      %let DROP_NUM = &DROP_NUM %qtrim( &&NUM&I ) ;
    %END;
  %end ;
RUN;

%END;

%IF &N_CHAR > 0 %THEN %DO;
DATA _NULL_;
  %do I = 1 %to &N_CHAR ;
    %IF "%qtrim(&&CHARMAX&I)" eq "" %THEN %DO;
      %let DROP_CHAR = &DROP_CHAR %qtrim( &&CHAR&I ) ;
    %END;
  %end ;
RUN;

%END;

/*=====*/
/* Create output dataset
/*=====*/

data &DSNOUT ;

  %if &DROP_CHAR ^= %then %str(DROP &DROP_CHAR ; ) ; /* drop char variables
that
  have only missing values */

  %if &DROP_NUM ^= %then %str(DROP &DROP_NUM ; ) ; /* drop num variables
that
  have only missing values */

set &DSNIN ;

  %if &DROP_CHAR ^= or &DROP_NUM ^= %then %do;

```

```
        %put /-----\ ;
        %put | Variables dropped are &DROP_CHAR &DROP_NUM | ;
        %put \-----/ ;
    %end;

    %if &DROP_CHAR = and &DROP_NUM = %then %do;
        %put /-----\ ;
        %put |           No variables are dropped | ;
        %put \-----/ ;
    %end;

run ;

%L9999:

%mend DROPMISS ;
```