Paper 212-25

Using SAS® Macro to Explore the Contents of a Data Set

Kinwah Fung, Toronto Sunnybrook Regional Cancer Centre, Ontario Canada.

ABSTRACT

The CONTENTS procedure prints the descriptions of the contents of a SAS data set and one of the variables given in the output data set is FORMAT. This paper presents a SAS Macro which lists the original values and the formatted values of variables in addition to the information given by PROC CONTENTS. This can help us in identifying the distinct values of the variables before we proceed to any data management procedures, and verifying the validity of the data while listing the contents of the data set.

INTRODUCTION

Listing the contents of a data set is the first step in conducting any data analysis. The CONTENTS procedure prints the descriptions of the contents of files including data sets from a SAS library. However, the distinct values of a data set's variables will not be listed unless we use PROC FREQ, REPORT, TABULATE, or some other SAS procedures. A list of the distinct values of variables enables us to identify any data invalidity before we go on.

This paper exhibits a SAS Macro that uses PROC SQL and data steps to generate a list of the original values and formatted values of variables in addition to the selected contents of a data set. A user just needs to specify 1) the library the data set lies, 2) the name of the data set, 3) the maximum number of distinct values of a variable to be shown, and 4) the page length of output.

This SAS Macro can be applied to any SAS data set from the WORK or any other specified libraries. It can contain formatted or unformatted variables. The type of the variables can be numeric or character.

If the number of distinct values of a variable exceeds the maximum number specified by the user, no list of the values will be printed for that variable. If the user knows beforehand that at least one variable has a long list of distinct values, the maximum number of distinct values to be shown can prevent from printing a huge amount of pages. For instance, there are 1000 patients with 988 distinct dates of birth in a data set. Setting the maximum number, let's say 50, can keep the printout from including the long list of dates of birth.

This SAS Macro uses the PUT statement to list the printout. The page length given by user will instruct the Macro when to advance a page and keep the headings in the right place.

DATA SET OF CONTENTS

The first step is to use PROC SQL to create a table storing six selected contents variables VARNUM, NAME, TYPE, LENGTH, LABEL, and FORMAT for the contents of the data set specified by the user. Including more contents variables requires slight modifications in the Macro.

MACRO VARIABLES TO BE CREATED

A Macro variable is created for the number of variables in the data set. This Macro variable NO_VAR controls the %DO loops to 1) create a set of Macro variables (NO_DV1, NO_DV2, …, NO_DV&NO_VAR) for the numbers of the variables' distinct values, 2) store distinct values of variables into Macro variables (P1_1, P1_2, …, P1_&NO_DV1, P2_1, P2_2, …, P2_&NO_DV2, …), and 3) to print the output list. Another six sets of Macro variables (VARN1-VARN&NO_VAR, NAME1-NAME&NO_VAR, TYPE1-TYPE&NO_VAR, LENG1-LENG&NO_VAR, LABE1-LABE&NO_VAR, and FORM1-FORM&NO_VAR) for the values of the six selected contents variables are generated by using data step statements and SYMPUT CALL routines.

DISTINCT VALUES OF VARIABLES

If the number of a variable's distinct values exceeds the maximum number specified by the user, the list of the distinct values will not be shown for that variable.

FORMATTED VALUES OF VARIABLES

If the variable is not formatted or is formatted with SAS formats, formatw.d (e.g. DATETIME16, DATE9, $8., or 10.), the list of original distinct values only will be printed. The formatted values following a '=' sign will be listed with the original values if the variable is formatted with a user-written format.

LIST OF THE CONTENTS

The user specifies the length of page. This SAS Macro uses the default for the line size 132. If the user wants to use different line size, s/he may need to modify the position in the PUT statement of the Macro. The heading will be repeated on every new page. If a list of a variable cannot be fitted into one page, a sign '(Cont.)' will be printed on the new page. The selected contents of a data set are position, variable name, distinct values, type, length, format, and the number of distinct values.

THE CODES

```
**************************************************************************
This %contents Macro generates the contents of a SAS dataset and
the distinct original and formatted values of the variables. The user
needs to specify the following 4 parameters:
LIBNAME - the LIBNAME where the data set
          lies.
DSNAME - the data set.
MAX_NV - the maximum number of distinct
          values to be shown.
PGLENGTH - the length of the page.
**************************************************************************;
%macro
contents(libname=,dsname=,max_nv=,pglength=);

title "%upcase(&libname..&dsname)";

/*Create an output dataset for the contents*/
proc sql;
create table varlist as
select varnum, name, type, length, label, format
from dictionary.columns
where libname=%upcase("&libname") and
   memname=%upcase("&dsname")order by varnum;
quit;

/* Number of variables in dataset */
proc sql noprint;
select max(varnum) into: no_var
from varlist;
quit;

/* Use the data step SYMPUT CALL routine to move
the values of variables VARNAME, NAME, TYPE,
LENGTH, LABEL, and FORMAT into Macro variables
*/
data _null_;
set varlist;
call
symput('varn'||left(put(_n_,4.)),compress(varnum
));
call
symput('name'||left(put(_n_,4.)),trim(name));
call
symput('type'||left(put(_n_,4.)),compress(upcase
(type)));
```

```
call
symput('leng'||left(put(_n_,4.)),compress(length
));
call
symput('labe'||left(put(_n_,4.)),trim(label));
call
symput('form'||left(put(_n_,4.)),trim(format));
run;

/* Store the distinct values of variables into
Macro variabes */
%do i=1 %to &no_var;
   proc sql;
   create table var&i as
   select distinct &&name&i as dv from
        &libname..&dsname;
   quit;
   proc sql noprint;
        select count(*) into: no_dv&i
        /* Number of distinct values */
        from var&i;
   quit;

   /* Match the original value with the
formatted value if there is any */
   %if &&no_dv&i <= &max_nv %then %do;
        data _null_;
        length temp_fmt $ 30;
        set var&i;
        temp_fmt="&&form&i";
        temp_lg=length("&&form&i");
        %if &&form&i ne  %then %do;
             if substr(temp_fmt,temp_lg-1,1) in
             ('1','2','3','4','5','6','7','8','9'
             ,'0') then do;
                %if &&type&i=NUM %then %do;
                   call
                   symput("d&i._"||left(put(_n_,3
                   .)),compress(put(dv,&&form&i..
                   )));
                %end;
                %else %do;
                   call
                   symput("d&i._"||left(put(_n_,3
                   .)),trim(put(dv,&&form&i..)));
                %end;
             end;
             else do;
                %if &&type&i=NUM %then %do;
                   call
                   symput("d&i._"||left(put(_n_,3
                   .)),compress(dv)||'='||trim(pu
                   t(dv,&&form&i..)));;
                %end;
                %else %do;
                   call
                   symput("d&i._"||left(put(_n_,3
                   .)),trim(dv)||'='||trim(put(dv
                   ,&&form&i..)));;
                %end;
             end;
        %end;
        %else %if &&form&i=  %then %do;
             %if &&type&i=NUM %then %do;
                call
                symput("d&i._"||left(put(_n_,3.))
                ,compress(dv));
             %end;
             %else %do;
                call
                symput("d&i._"||left(put(_n_,3.))
                ,trim(dv));
             %end;
        %end;
        run;
   %end;
   proc sql;
   drop table var&i;
   quit;
%end;

/* Write the list into output file */
data _null_;
file print;
%do i=1 %to &no_var;
   if &i=1 | &pglength-line_no<=0 then do;
        line_no=1;
        put _page_ #line_no @1 'Position' @10
        'Variable' @19 'Values' @61 'Type' @66
        'Length' @73 'Label' @115 'Format' @125
        'Distinct';
         line_no=line_no+2;
   end;
   %if &&no_dv&i > &max_nv %then %do;
        put #line_no @1 "&&varn&i" @10 "&&name&i"
        @19 'Not to Show' @61 "&&type&i" @66
        "&&leng&i" @73 "&&labe&i" @115 "&&form&i"
        @125 "%cmpres(&&no_dv&i)";
        line_no=line_no+1;
   %end;
   %else %do j=1 %to &&no_dv&i;
        if &pglength-line_no<=0 then do;
             line_no=1;
             put _page_ #line_no @1 'Position'
             @10 'Variable' @19 'Values' @61
             'Type' @66 'Length' @73 'Label' @115
             'Format' @125 'Distinct';
             line_no=line_no+2;
        end;
        %if &j=1 %then %do;
             put #line_no @1 "&&varn&i" @10
             "&&name&i" @19 "&&d&i._&j" @61
             "&&type&i" @66 "&&leng&i" @73
             "&&labe&i" @115 "&&form&i" @125
             "%cmpres(&&no_dv&i)";
             line_no=line_no+1;
        %end;
        %else %do;
             if line_no=3 then do;
                put #line_no @1 "&&varn&i
                (cont.)" @19 "&&d&i._&j";
                line_no=line_no+1;
             end;
             else do;
                put #line_no @19 "&&d&i._&j" ;
                line_no=line_no+1;
             end;
        %end;
   %end;
   put ' ' ;
   line_no=line_no+1;
%end;
run;
%mend td;
```

AN EXAMPLE

Here is an example for the use of this SAS Macro. Table 1 shows the partial printout of a PROC CONTENTS and Table 2 is the printout from the Macro listing the distinct values of the variables.

```
%contents(libname=work,dsname=pt_info,max_nv=50,pglength=46)
;
```

PRINTED OUTPUT

The SAS Macro prints the following:

1. the title with the LIBNAME and the name of the data set.
2. the variable number in the data set.
3. the variable name.
4. 'Not to Show' if the number of distinct values exceeds the maximum.
5. the distinct values of the variable and the formatted values following a '=' sign if the number of distinct values does not exceed the maximum number specified by the user.
6. the type of the variable, character or numeric.
7. the length of the variable.
8. the label if any.

9. the format.
10. the number of distinct values including missing value of the variable.

CONCLUSION

This SAS Macro lists the distinct values of variables in addition to some selected PROC CONTENTS variables. The list enables the users to browse the details of a data set and identify any invalidity before using other procedures to describe the data set. It simplifies the tasks of data managers and analysts.

REFERENCES

SAS Institute Inc., SAS Procedures Guide, Version 6, Third Edition, 1990. SAS Institute Inc., Cary, NC.

SAS Institute Inc., SAS Macro Language: Reference, First Edition, 1990. SAS Institute Inc., Cary, NC.

SAS Institute Inc., SAS Guide to the SQL Procedure: Usage and Reference Version 6, First Edition, 1989. SAS Institute Inc., Cary, NC.

SAS Institute Inc., SAS Sample Programs from Technical Support http://ftp.sas.com/techsup/download/sample/unix/dstep/csv_labels.sassrc.

CONTACT INFORMATION

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

Kinwah Fung
Department of Clinical Trials and Epidemiology,
Toronto Sunnybrook Regional Cancer Centre,
2075 Bayview Avenue
Toronto, Ontario M4N 3M5 Canada
Email: kinwah.fung@tsrcc.on.ca

Table 1. Partial Printout of PROC CONTENTS

#	Variable	Type	Len	Pos	Format	Informat	Label
1	CHART	Num	8	0	6.	7.	Patient Chart #
2	DOB	Num	8	8	DATE9.	DATE9.	Date of Birth
6	INSERT_D	Num	8	31	DATE9.	DATE9.	Date of Insert
5	PT_GROUP	Num	8	30	1.		Patient group
3	SEX	Char	6	16	$GENDER.	$1.	Gender
4	TYPE_CA	Num	8	22	TYPE_CA.	3.	Type of Cancer

Table 2. Printout generated by the SAS Macro

① MYLIB.PT_INFO

② #	③ Variable	④ Values	⑥ Type	⑦ Length	⑧ Label	⑨ Format	⑩ Distinct
1	CHART	Not to Show	NUM	8	Patient Chart #	6.	232
2	DOB	Not to Show	NUM	8	Date of Birth	DATE9.	230
3	SEX	⑤ 0=Female 1=Male F=Female M=Male	CHAR	6	Gender	$GENDER.	4
4	TYPE_CA	1=Breast 2=Prostate 3=Lung 4=Head & Neck 5=Bladder 6=Esophagus 7=Pancreas 9=Renal Cell/Kidney 12=Colorectal 13=Anal Canal 14=Multiple Myeloma 15=Unknown 16=Other	NUM	8	Type of Cancer	TYPE_CA.	13
5	PT_GROUP	1 2 3	NUM	8	Patient group	1.	3
6	INSERT_D	13MAY1999 14MAY1999 15MAY1999 16MAY1999 17MAY1999	NUM	8	Date of Insert	DATE9.	5