

Paper 032-30

A SAS[®] Macro for Producing Customized Reports

Lan Zhao, Yajie Wang and Bob Edson
VA Palo Alto Health Care System, Menlo Park, CA

ABSTRACT

To produce highly customized reports, we use DATA _Null_ and PUT statements. In the process, we often need to consider the unique requirements and specifications for a given report. Those requirements and specifications include but are not limited to the following: column positions and headings, row label width and indentation, blank lines between rows, number of columns that appear on each page, page breaks, and other design specifications. The work quickly becomes tedious, repetitive, and time consuming.

A question naturally arose from this process, namely, can we generalize the above procedure by creating a SAS macro to accomplish the objective? This paper presents a SAS macro that aims to provide a solution to this problem. We also demonstrate the construction of such a macro using a concrete example.

INTRODUCTION

This section demonstrates some key properties of the SAS macro Sugi30Dsbreport that we have developed. To explain all the steps clearly, we demonstrate through an example. Let us start with the following sample SAS dataset called "final" for which we want to create a report. The six columns represent the six hospitals or study sites, and the rows represent various summary statistics (counts, mean and standard deviation, and range) for age and ethnicity by individual hospital. Notice that there are neither row labels nor column headings for the dataset. These are added and report formatting is done in the following steps.

Obs	COL1	COL2	COL3	COL4	COL5	COL6
1	55	55	50	55	52	83
2	51	55	50	55	52	83
3	41.4 (6.4)	41.6 (7.6)	42.5 (6.6)	43.8 (7.0)	41.6 (6.7)	43.2 (7.1)
4	25 - 53	27 - 58	21 - 53	31 - 62	30 - 55	28 - 63
5	7	7	10	8	12	6
6	7	10	5	7	7	10
7	4	2	7	6	5	10
8	9	8	4	5	3	11
9	12	5	5	8	4	13
10	2	8	5	6	4	11
11	4	5	4	9	9	9
12	10	10	10	6	8	13

Step 1: Label the rows, indent the row labels, produce desired blank lines between rows and indicate page break line for the dataset according to one's design and specification by creating a row label dataset called "rlabels".

```

data rlabels;
  infile datalines;
  input breakpage 1-1 datanext 3-3 @7 labl $char50.;

  /**
   breakpage =1 indicates that at current line, the page will break.
   Otherwise, there will be no break.

   datanext =1 indicates that the corresponding data that we started with
   will appear next to the row label; otherwise no data will appear there.

   See the following output report for details.
  ***/

datalines;
  1   Randomized

      Age
  1   N
  1   Mean (Std)
  1   Min, Max

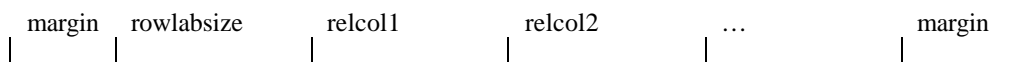
  1   Ethnicity
  1   White, not Hispanic
  1   Black, not Hispanic
  1   Hispanic, White
  1   Hispanic, Black
  1   American Indian
  1   Pacific Islander
  1   Asian
  1   Other
;
run;

```

If we want more blank lines between rows, we simply insert them into the rlabels data file above.

Step 2: Design the layout of the report by giving column positions, number of columns per page, column headings, margins, row label width, titles, footnotes, etc.

Assume we want the report to have the following layout:



We specify the margin (same for left and right so the table is centered on the page), row label width and column positions as follows:

```

%let linesize=89; /** line size = 89 */
%let margin=4; /** left and right margins are same */
%let rowlabsize=25; /** row label width */

/** if we want to have a maximum of four columns appear on one page */
/** (that is, colperpage=4), */
/** then we need to specify four starting positions relative to where */
/** the row label ends as follows */

%let relcol1=0; /** 1st column starts from where row label ends */
%let relcol2=12; /** 2nd column starts 12 positions to right of where row label ends */
%let relcol3=28; /** 3rd column starts 28 positions to right */
%let relcol4=42; /** 4th column starts 42 positions to right */

```

```
%let title1=%nrstr (Study Name);
%let title2=%nrstr (General Baseline Characteristics);
%let title3=%nrstr (of Randomized Patients);
```

```
%let foot1=%nrstr(note: any notes);
```

Now apply the SAS macro Sugi30Dsmreport:

```
%Sugi30Dsmreport(inset=final, rlabeldata=rlabelds,
  collabel1=%nrstr(" Hsp1", " Hsp2", " Hsp3", " Hsp4",
    " Hsp5", " Hsp6"),
  collabel2=%nrstr("(701)", "(703)", "(708)", "(712)",
    "(716)", "(723)"),
  collabelalign=%str(-L),
  coltot=6, colperpage=4,
  align=%nrstr(-L),
  outtbl=sugi30,
  titlnum=3, footnum=1)
```

The file "sugi30" appears as follows:

Study Name				
General Baseline Characteristics				
of Randomized Patients				
	Hsp1 (701)	Hsp2 (703)	Hsp3 (708)	Hsp4 (712)

Randomized	55	55	50	55
Age				
N	51	55	50	55
Mean (Std)	41.4 (6.4)	41.6 (7.6)	42.5 (6.6)	43.8 (7.0)
Min, Max	25 - 53	27 - 58	21 - 53	31 - 62

note: any notes				

page 2

Study Name
General Baseline Characteristics
of Randomized Patients

	Hsp1 (701)	Hsp2 (703)	Hsp3 (708)	Hsp4 (712)

Ethnicity				
White, not Hispanic	7	7	10	8
Black, not Hispanic	7	10	5	7
Hispanic, White	4	2	7	6
Hispanic, Black	9	8	4	5
American Indian	12	5	5	8
Pacific Islander	2	8	5	6
Asian	4	5	4	9
Other	10	10	10	6

note: any notes

page 3

Study Name
General Baseline Characteristics
of Randomized Patients

	Hsp5 (716)	Hsp6 (723)

Randomized	52	83
Age		
N	52	83
Mean (Std)	41.6 (6.7)	43.2 (7.1)
Min, Max	30 - 55	28 - 63

note: any notes

page 4

Study Name
General Baseline Characteristics
of Randomized Patients

	Hsp5 (716)	Hsp6 (723)

Ethnicity		
White, not Hispanic	12	6
Black, not Hispanic	7	10
Hispanic, White	5	10
Hispanic, Black	3	11
American Indian	4	13
Pacific Islander	4	11
Asian	9	9
Other	8	13

note: any notes

KEY PORTION OF THE MACRO

- (1) Combine the starting dataset “final” with the row label dataset “rlabels”, so that the combined dataset has the desired row label and spaces.**

In our example, the combined dataset appears as follows.

Obs	breakpage	datanext	labl	COL1	COL2	COL3	COL4	COL5	COL6
1	.	1	Randomized	55	55	50	55	52	83
2	.	.							
3	.	.	Age						
4	.	1	N	51	55	50	55	52	83
5	.	1	Mean (Std)	41.4 (6.4)	41.6 (7.6)	42.5 (6.6)	43.8 (7.0)	41.6 (6.7)	43.2 (7.1)
6	.	1	Min, Max	25 - 53	27 - 58	21 - 53	31 - 62	30 - 55	28 - 63
7	.	.							
8	1	.	Ethnicity						
9	.	1	White, not Hispanic	7	7	10	8	12	6
10	.	1	Black, not Hispanic	7	10	5	7	7	10
11	.	1	Hispanic, White	4	2	7	6	5	10
12	.	1	Hispanic, Black	9	8	4	5	3	11
13	.	1	American Indian	12	5	5	8	4	13
14	.	1	Pacific Islander	2	8	5	6	4	11
15	.	1	Asian	4	5	4	9	9	9
16	.	1	Other	10	10	10	6	8	13

The following is the part of SAS macro Sugi30Dsmreport for combining datasets as described above.

```
data &rlabeldata;  /*** &rlabeldata takes the row label dataset “rlabels” in our example ***/
  set &rlabeldata;
  order=_n_;
run;
```

```

proc sort data=&rlabeldata ;
  by order;
run;

data rowlabelselec;
  set &rlabeldata ;
  where datanext=1;
run;

data combi;
  merge rowlabelselec &inset;  /** &inset takes the dataset "final" in our example **/
run;

proc sort data=combi;
  by order;
run;

data final ;
  merge &rlabeldata combi;
  by order;
run;

```

(2) Generate a report that breaks up the dataset into column groups

The dataset produced by (1) above almost always has more columns and/or rows than will fit on one page, so we need a way to break up the dataset into pieces where each piece appears on a separate page. In Step 1 we discussed how to break up the dataset by rows (using breakpage). Here, we describe how to split the dataset into column groups so that each column group starts on a different page (or set of pages when the dataset has more rows than can fit on one page). Then, in combination with the repeated use of a simple SAS macro called OneReport (discussed in (3) below) both inside and outside the do-loop, we perform the complex task of creating the report by performing a simple task multiple times .

The following part of the SAS macro Sugi30Dsmbreport sets up the dataset for use of OneReport. Given the total number of columns (coltot) and the maximum number of columns in a group (colperpage), the macro calculates the number of full column groups (denoted numfullcolgrps) which is the integer portion of $G = \text{coltot} / \text{colperpage}$ and processes these inside the do loop. If G has a remainder, then the report for the last column group has fewer columns than colperpage, and the macro automatically centers the output.

```

/** finding the number of full column groups */
/** For our example, a full group has four columns */

%let numfullcolgrps=%sysevalf(&coltot/&colperpage, floor);

%let border=%eval(&numfullcolgrps * &colperpage);

%do i=1 %to &border %by &colperpage;

  %let endpos=%eval(&i + &colperpage -1);

  /**
    Sometimes the last column of the report, which usually shows the values across all
    hospitals, needs to be wider than the rest of columns, so we add some extra spaces by
    specifying collastlenadd in the OneReport. For details on the parameters for OneReport,
    please check key portion (3) below.
  */

  %OneReport(startpos=&i, endpos=&endpos, adjpos=0, collastlenadd=0)

%end;

%if &border~=&coltot %then %do;

```

```

/**
    Because the last column group in our example has fewer than colperpage columns, we need
    to adjust the margins so that the output will be centered.
    To do this, we determine the value of colminus, defined as colperpage minus the number of
    columns in the last group. Then, we calculate the number of spaces to add to each margin
    (called adjpos) by determining the number of positions represented by the first colminus
    columns in a full column group and dividing that by 2. In our example, colminus = 4-2 or 2, and
    since the first two columns contain 28 positions (see Step 2 above), 14 spaces are added to
    each margin.
**/

%let colminus=%eval(&colperpage-(&coltot-&border));

%let adjpos=0;

%do i=1 %to &colminus;
    %local wid&i;
    %let iplus1=%eval(&i+1);

    %let wid&i=%eval(&&relcol&iplus1-&&relcol&i);  /** column width **/
    %let adjpos=%eval(&adjpos+&&wid&i);
%end;

%let adjpos=%eval(&adjpos-&collastlenadd);

/** adjpos is the extra blank spaces added to one side margin **/

%let adjpos=%sysevalf(&adjpos/2, floor);
%OneReport(startpos=%eval(&border+1),
           endpos=&coltot, adjpos=&adjpos, collastlenadd=&collastlenadd)
%end;

```

(3) Establish a SAS help macro OneReport to generate the report for a single column group.

Parameters for OneReport:

startpos=	column number for first column on a given page
endpos=	column number for last column on a given page
adjpos=	extra blank spaces added to one side margin when last column group of report has fewer columns than previous column groups
collastlenadd=	extra spaces added to the last column of the report

In our example, the values of startpos and endpos are 1 and 4 for pages 1-2, and 5 and 6 for pages 3-4.

The following is the main part of the SAS macro OneReport.

```

data _null_;
    set final (keep=labl col&startpos-col&endpos breakpage datanext)
           end=eoj;  /** the dataset final is a combined dataset shown in (1) **/

    file "&outtbl" print notitles linesleft=remain n=ps mod;

    array aryhead1[&coltot] $100 head1_1-head1_&coltot (&collabel1);

    %if &collabelnum=2 %then %do;
        array aryhead2[&coltot] $100 head2_1-head2_&coltot (&collabel2);
    %end;

```

```

retain LINE;

LINE=repeat("-", &linesize.-2*&margin. -2*&adjpos);

if _n_=1 or breakpage then do; /** first page or page break */
    link newpage;
end;

put                                /** add row labels */
    @(&margin + &adjpos + 1) labl $char&rowlabsize..
    %do i=&startpos %to &endposmins1;
        %let j=%eval(&i-&startpos+1);
    /**
        j starts from the 1st column to the column right before the last column on a given page,
        in our example, j starts from 1 to 3 for pages 1-2, and 1 for pages 3-4.
    */
    /** add data to columns */
        @(&margin + &adjpos + &rowlabsize + &&relcol&j) col&i
            $char&&rellen&j... &align /** &&rellen&j is the width for the jth column */
            /** that is &&relcol&jplus1 - &&relcol&j */
        %end;

    %let j=%eval(&endpos-&startpos+1);
    /**
        now j is the last column on a given page,
        in our example, j takes value 4 for pages 1-2, and 2 for pages 3-4.
    */
    /**
        sometimes the last column is wider than the
        rest, so we output it separately.
    */

    @(&margin + &adjpos + &rowlabsize + &&relcol&j)
        col&endpos $char&collastlen.. &align
        ;

if breakpage or eoj then do;
    put                                /** add line above foot notes */
        @(&margin + &adjpos + 1) LINE//
        %do i=1 %to &footnum;

            @(&margin + &adjpos + 1) "&&foot&i"/

        %end;
        ;
end;

return;
newpage:
put _page_//

    %do i=1 %to &titlnum;            /** add titles */
        @&&titl&i.pos "&&title&i" /
    %end;
    //

    %do i=&startpos %to &endposmins1; /** add column headings */
        %let j=%eval(&i-&startpos+1);
        @(&margin + &adjpos + &rowlabsize + &&relcol&j) head1_&i
            $char&&rellen&j... &collabelalign
    %end;

```



```

%let j=%eval(&endpos-&startpos+1);

/**
  sometimes the last column is wider than the
  rest, so we output it separately.
**/

@(&margin + &adjpos + &rowlabsize + &&relcol&j)
  head1_&endpos $char&collastlen.. &collabelalign
/

%if &collabelnum=2 %then %do;
  %do i=&startpos %to &endposmins1;
    %let j=%eval(&i-&startpos+1);
    @(&margin + &adjpos + &rowlabsize + &&relcol&j) head2_&i
      $char&&rellen&j... &collabelalign
  %end;

  %let j=%eval(&endpos-&startpos+1);

/**
  sometimes the last column is wider than the
  rest, so we output it separately.
**/

  @(&margin + &adjpos + &rowlabsize + &&relcol&j)
    head2_&endpos $char&collastlen.. &collabelalign
/** &collastlen is the width for the last column of the report **/

  /
  %end;
  @(&margin + &adjpos + 1) LINE;
%end;

run;
%mend OneReport;

```

CONCLUSION

In this paper, we demonstrated the construction of a SAS macro that can be efficiently used in addressing the needs for various output format requirements and specifications. In addition, we demonstrated the specific values used to complete a complex task, producing a multi-page report, by performing a set of simple tasks. In our example, we split the dataset to appear on four pages using a relatively simple SAS macro. These steps made the overall task of creating and modifying the program much easier and more enjoyable.

APPENDIX

Macro Name: Sugi30Dsmbreport

PURPOSE:

Generate Data and Safety Monitoring Board (DSMB) report given inset and rlabldata set

PARAMETERS:

inset= dataset containing data to produce report, without any labels

rlabldata= dataset containing row label dataset, which can be edited according to the desired appearance of the report.

collabel1= 1st column heading string, which usually is a site or treatment name

collabel2= 2nd column heading string

collabelnum= number of column headings

collabelalign= indicator for left, right or center alignment of column heading, and it takes values %str(-R), %str(-L), %str(-C)

coltot= total number of columns in inset

colperpage= maximum number of columns that appear on one page

align= indicator for left, right or center alignment of data, and it takes value %str(-L), %str(-R) or %str(-C)

outtbl= file name for DSMB report

titlnum= total number of titles in the report

footnum= total number of foot notes in the report

collastlenadd= additional spaces added to the last column in the report, which takes any non-negative integers with default value set at 0.

```
%macro Sugi30Dsmbreport(inset=, rlabeldata=, collabel1=%quote( ),
    collabel2=%str(),
    collabelnum=2,
    collabelalign=%str(-C),
    coltot=, colperpage=, align=%nrstr( ),
    collastlenadd=0,
    outtbl=, titlnum=, footnum=)
```

REFERENCE

SAS Institute Inc. (1997), SAS Macro Language: Reference, Version 6, Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

This work was supported by the Department of Veterans Affairs Cooperative Studies Program and conducted under the guidance of Philip Lavori, Ph.D., director of the Palo Alto Cooperative Studies Program Coordinating Center. We also would like to express our appreciation to the following people for their contributions on this project: Surai Thaneemit-Chen, Vaishali Krishnan, Galina Shamayeva, Lauren Uyeda and Kristi Mahadocon.

CONTACT INFORMATION

Ms. Lan Zhao
 VA Palo Alto Health Care System
 CSPCC (151K)
 795 Willow Road
 Menlo Park, CA 94025
 Work Phone: 650-493-5000 ext 23690
 Email: lan.zhao@med.va.gov

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 registered trademarks or trademarks of their respective companies.