

Automating the Number of Treatment Columns for a Dose Escalation Study

Catherine DeVerter, Novella Clinical, Morrisville, NC
 Sonali Garg, Alexion Pharmaceuticals, Cheshire, CT

ABSTRACT

In dose escalation studies, there is the need to increase or decrease the number of treatment columns in table output when a new treatment arm is added to the trial. If programs are not written to add the new columns dynamically, they often need to be edited repeatedly. What if we can automate the number of columns to be displayed in the table without having to go and edit the program?

This paper will show how SAS programmers can automate and control the number of columns in the table output without having to change the PROC REPORT's COLUMN and DEFINE statements again and again. The method discussed in this paper will make use of the SQL procedure.

INTRODUCTION

In some early phase clinical trials (such as Phase I oncology), it is common to perform dose escalation studies to find the most suitable dose for the drug being evaluated. Often a new dose is added as the trial progresses and outputting all the columns for table output may turn into a nightmare if the change has to be done manually each time. To further complicate things, sometimes the new dose is calculated as existing doses are evaluated (e.g., based on PK results), so the exact doses evaluated in the trial may be not be known until statistical programming has already begun.

Macros can be written to have a treatment column outputted automatically for any new dose group that has been entered into the clinical database so that TLF creation is automated.

DETERMINE TREATMENT VARIABLES WITH THE %TRTCOLS MACRO

The %TrtCols macro reads from the key subject level dataset (e.g., ADSL) to determine the different values that currently exist for the treatment variable. The numeric treatment variable TRT01PN may be used, or any other variable can be created that indicates the unique dose. For the purpose of this paper, we will use TRT01PN.

	SUBJID	TRT01P	TRT01PN	ITTFL	SEX	AGE
1	01-101	10 mg	1	Y	M	25
2	01-102	10 mg	1	Y	M	36
3	01-103	50 mg	3	Y	F	41
4	01-103	50 mg	3	Y	M	35
5	01-104	25 mg	2	Y	F	22
6	01-106	25 mg	2	Y	F	50
7	01-108	10 mg	1	Y	M	48
8	01-110	25 mg	2	N	M	32

Table 1. Subject Level Dataset

The macro assumes that you have created a dataset called DENOM that has only one record and numeric variables representing the N (i.e., number of subjects) of each treatment group. DENOM is a PROC FREQ of the treatment variables of your subject level dataset that is then transposed with the new variables having a prefix of 'COLUMN_'. A variable for the overall total (or MTD/Maximum Tolerated Dose, if needed) can also be included. Table 2 displays the DENOM dataset that results from the subject level dataset created for our example, restricted to subjects in the ITT analysis population:

COLUMN_1	COLUMN_2	COLUMN_3	COLUMN_99
3	2	2	7

Table 2. DENOM Dataset

Note that DENOM contains one COLUMN_x variable for each value of TRT01PN along with COLUMN_99, which represents the overall Total (i.e., All Subjects).

INSIDE THE %TRTCOLS MACRO

%TrtCols uses the distinct values of TRT01PN to create a global macro variable COLS that is a string of variables COLUMN_1-COLUMN_X along with a variable for the overall Total and/or MTD, if needed. Here, X denotes the value of the assigned treatment. The resulting COLS variable will be included in the COLUMNS statement of PROC REPORT.

The parameters used by %TrtCols are POP, to indicate the desired analysis population, and ALL, to designate a value to use for the overall Total column.

```

%macro TrtCols(pop=ITTF, all=99) ;

%global trttotcol cdesc nlist cols ;

/* Create dataset with analysis population counts for each treatment and the Total;
data adsl_ ;
  set adata.adsl (where=(&pop='Y' and trt01pn>.z)) ;
  output ;
  %if &all^= %then %do ;
  * Output for overall total ;
  trt01pn=&all ;
  output ;
  %end ;
run ;

proc sql noprint nowarn ;
  *** CDESC - Treatment labels (e.g., 10 mg, 20 mg, 50 mg) ;
  *** COLS - List of treatment column variables (e.g., COLUMN_1) ;
  *** NLIST - Number of treatment groups, including Total ;

  select distinct
    /* Get distinct values of TRT01P for treatment column labels */
    case
      when trt01pn=&all then 'Total'
      else trt01p
    end into :cdesc separated by ":"
  from adsl_
  order by trt01pn ;

  select name into :cols separated by " "
  from dictionary.columns
  where libname='WORK' and memname="DENOM" and name like 'COLUMN%' ;

  %let nlist=%sysfunc(countw(&cdesc,':')) ;
quit ;

/* Using DENOM, string the values together in the macro variable TRTN ;
data _null_ ;
  set denom ;
  length trtlist $100 ;
  retain trtlist ' ' ;
  %do i=1 %to &nlist ;
  %let vname=%scan(&cols,&i,%str( )) ;
  trtlist=catx(':', trtlist, strip(&vname)) ;
  %if &i=&nlist %then call symput('trtn', trtlist) ;
  %end ;
run ;

/* Cycle through trt variables and assign to TRTxCOL macro variables, ;
/*   where x=1, 2, ..., number of treatments. TRTTOTCOL is for the Total column. ;
/* These are the treatment column headers. '|' is assumed to be the SPLIT ;
/* character used in PROC REPORT ;
%do z=1 %to &nlist ;
  %let trtlbl=%scan(&cdesc,&z,:) ;

```

```

%let trtpopn=%scan(&trtn,&z,:);
%if %scan(&trtlbl,1)^=Total %then %do;
  %global trt&z.col;
  %let trt&z.col=&trtlbl|(N=&trtpopn);
%end;
%else %let trttotcol=&trtlbl|(N=&trtpopn);
%end;

%mend TrtCols;

```

The macro variables created are displayed in Output 1.

```

CDESC=10 mg:25 mg:50 mg:Total
COLS=COLUMN_1 COLUMN_2 COLUMN_3 COLUMN_99
NLIST=4

TRTN=3:2:2:7

TRT1COL=10 mg|(N=3)
TRT2COL=25 mg|(N=2)
TRT3COL=50 mg|(N=2)
TRTTOTCOL=Total|(N=7)

```

Output 1. Macro Variables Created from %TrtCols

GENERATING DEFINE STATEMENTS WITH THE %DEFSTMT MACRO

The %DefStmt macro dynamically creates the DEFINE statements that will be used in PROC REPORT for your table output. It uses the SASHELP VCOLUMN view with PROC SQL to get the treatment variable names (COLUMN_1, COLUMN_2, etc.) and references the COLS macro variable created in the %TrtCols macro.

INSIDE THE %DEFSTMT MACRO

%DefStmt assumes that the dataset containing your table data, represented by the macro parameter DSNAME, has already been created and has the treatment columns in variables named COLUMN_1, COLUMN_2, etc. The other parameters used by %DefStmt are JUST, to indicate justification for the column data values, and WIDTH, for the column width.

The end result of %DefStmt is a global macro variable called DEFSTMT which is a string of DEFINE statements for each treatment column.

```

%macro defStmt(dsname=, just=1, width=);
%global defstmt;

data &dsname.0 (keep=&cols);
  set &dsname;
run;

data &dsname.1 (KEEP=col name);
  set sashelp.vcolumn (where=(memname="%upcase(&dsname.0)" )) end=eof;
  length col $3;

  col=strip(put(_n_,best.));
  if eof then col="tot";
run;

proc sql noprint;
  /* DEFINE statements for PROC REPORT */
  select 'define ' || strip(name) || ' / %nrstr("%")&trt' || strip(col)
  || 'col%nrstr("%") style(column)=[just=&just. cellwidth=&width]'
  into :defstmt separated by '; '
  from &dsname.1;

```

```
quit ;
%let defstmt =&defstmt%nrstr(;;

%mend defstmt ;
```

The value of the resulting global DEFSTMT macro variable is displayed in Output 2.

```
define COLUMN_1 / "10 mg|(N=3)" style(column)=[just=1 cellwidth=12%]; define
COLUMN_2 / "25 mg|(N=2)" style(column)=[just=1
cellwidth=12%]; define COLUMN_3 / "50 mg|(N=2)" style(column)=[just=1
cellwidth=12%]; define COLUMN_99 / "Total|(N=7)"
style(column)=[just=1 cellwidth=12%];
```

Output 2. DEFINE statements created from %DefStmt

CALLING THE %TRTCOLS AND %DEFSTMT MACROS

The %TrtCols macro must be called prior to the %DefStmt macro, and both are called in the table program prior to the PROC REPORT statement. The DEFSTMT macro variable created by %DefStmt is referenced inside PROC REPORT, as shown below.

```
%TrtCols(pop=ITTFLL, all=99) ;
%DefStmt(dsname=tabdata, width=12%) ;

ods rtf file="&out\Tables\T02.rtf" style=myStyle notoc_data ;
proc report data=tables.tabdata nowindows split='|' missing
style(report)=[width=100% cellpadding=0]
style(column)=[protectspecialchars=off asis=on just=1]
style(header)=[asis=on just=c] ;
column rowlbl &cols ;

define rowlbl / "Parameter" style(column)=[width=15%] style(header)=[just=1] ;
%unquote(&defstmt) ;
run ;
ods rtf close ;
```

Finally, Display 1 demonstrates the output of the simple table created above using %TrtCols and %DefStmt.

XYZ Pharma, Protocol: XYZ-0021				Page 1 of 1
Table 2 Demographics Summary All Subjects				
Parameter	10 mg (N=3)	25 mg (N=2)	50 mg (N=2)	Total (N=7)
Gender				
Female	0	2 (100.0%)	1 (50.0%)	3 (42.9%)
Male	3 (100.0%)	0	1 (50.0%)	4 (57.1%)

Display 1. Table Output using Dynamically Created Treatment Column Headers

LIMITATIONS/ASSUMPTIONS

- The code presented here works as is unless there are more than 5 or 6 treatment columns to display, depending on column width. When there are more than that, formatting might behave unpredictably depending on the ODS destination and options used. The example we have shown uses ODS RTF.
- %TrtCols and %DefStmt have to be called before PROC REPORT.
- The DEFINE statements and the treatment variables in the COLUMN statement of PROC REPORT will be listed in the order corresponding to the TRT01PN value.

CONCLUSION

This paper demonstrates how we can automate the display of treatment columns in PROC REPORT when the number of treatments is unknown or the actual treatment values are assigned during the course of the trial. The method discussed takes advantage of SASHELP views, PROC SQL and the SAS Macro Language, and allows for programming that will adapt dynamically to changing treatment groups without having to manually update the TLF program whenever a new treatment group is added.

CONTACT INFORMATION

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

Name: Catherine DeVerter
Enterprise: Novella Clinical
Address: 1700 Perimeter Park Drive
City, State ZIP: Morrisville, NC 27560
E-mail: cdeverter@novellaclinical.com

Name: Sonali Garg
Enterprise: Alexion Pharmaceuticals
Address: 352 Knotter Drive
City, State ZIP: Cheshire, CT 06410
E-mail: gargs@alxn.com

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.