Methods for Flexibly Controlling Combined and Customized Outputs
Xiaoyu (Sean) Liu, Novartis Oncology

ABSTRACT
When we make reports, tables and listings by using SAS PROC REPORT, PROC TABULATE, DATA _NULL_ and other statistical procedures, etc., sometimes we need to combine them together. For instance, patient profiles consist of many modules, such as demography, adverse events, ECG, lab. results and vital signs, etc. We need to combine them and flexibly to control their outputs based on the different format requirements. Sometimes we want to insert modules with page break and sometimes without page break. The purposes of this paper are: (1) to present some general examples for combination; (2) to offer some methods and options for controlling combined outputs. The options include FORMDLIM='delimiting-character', STARTPAGE=YES/NO in ODS, and _PAGE_ in COMPUTE block, and so on. This presentation is based on SAS version 8.2.

KEY WORDS: SAS, COMBINED OUTPUTS, METHODS, OPTIONS

INTRODUCTION
One of the major SAS programming work is to create tables, listings and reports. The complex tables/reports usually consists of multiple parts. We need to combine all the values from different sources and procedures. This paper introduced some methods to combine various outputs from different sources and procedures based on the requirements and formats. The three combination types are listed as bellows.

COMBINING OUTPUTS FROM PROC REPORT, PROC TABULATE AND DATA _NULL_ TOGETHER
We may use options FORMDLIM='delimiting-character' to combine the outputs from DATA _NULL_ and PROC REPORT. It specifies a character to delimit page breaks in SAS output. When the delimit character is null, a new physical page starts whenever a new page occurs. However, you can conserve paper by allowing multiple pages of output to appear on the same page. For example, this statement writes a line of dashes (- -) where normally a page break would occur:

Example 1: options formdlim=' ';
Title1 ;
TITLE2 "PROFILE OF AE LISTING BY CASE";
TITLE3 "CASE NUMBER: &casenum.";

data _null_;
COMBINE ODS OUTPUTS BY USING STARTPAGE=NO OR STARTPAGE=YES

Patient profiles consist of many modules. If we don’t need the page break between the listings, ods rtf startpage=no can be used. However, when we combine the patient profiles for multiple subjects, we need page break between subjects. Then we may use ods rtf startpage=yes.

Example 2

ods rtf startpage=no;
%do pi=1 %to &numpid ; %* start of loop for current patient ;

   title "Patient (&&pid&pi) Profile of Study &study ";
   title "Patient Profile(&&pid&pi) of Study &study (based on data extracted on &entydate.)";

   %do mi=1 %to &nummod; %* start of loop for current report module ;
   %let curmod=%scan(&mods,&mi,%str( ));
   %let curmod=rpt_&curmod;
   %if &mi>1 %then %do;
       ods rtf startpage=no;
COMBINING DERIVED COMPUTATION AND NOTES TOGETHER

When we combine the tables and listings together, we need to add some computation and footnotes to a single table/listing, but not all tables/listings. We can’t use FOOTNOTE statement. Compute block can be used to add them.

Example 3 The following codes can be used to make footnotes for adverse events in patient profiles.

```sas
compute after _page_ / style = {just=l asis=on};
  line @1 'Action taken:    0=No action taken';
  line @1 '                 1=Study drug dosage adjusted/temporarily interrupted';
  line @1 '                 2=Study drug permanently discontinued due to this ae';
  line @1 '                 3=Concomitant medication taken';
  line @1 '                 4=Non-drug therapy given';
  line @1 '                 5=Hospitalization/prolonged hospitalization';
endcomp;

_PAGE_ causes the compute block to execute once for each page, either immediately after printing any titles or immediately before printing any footnotes.

Example 4 We may use compute block to calculate % change from baseline and % change from min. for tumor size.

```sas
proc report data=tumor missing nowindows split = '#';
column  evl1n  les1n tumsize bsline change minchange;
define  evl1n /group   'Evaluation#No.' center
  style(column) = {cellwidth =  30 pt} style(header) = [just=c];
define  les1n/display order=internal 'Lesion#No.' center
  style(column) = {cellwidth =  35 pt} style(header) = [just=c];
define  tumsize/ analysis sum min 'Longest#diameter#for lesion' center
  style(column) = {cellwidth =  40 pt} style(header) = [just=c];
define  bsline/ 'Baseline' center
  style(column) = {cellwidth =  40 pt} style(header) = [just=c];
define change / computed “% change #from baseline” center
  style(column) = {cellwidth =  50 pt} style(header) = [just=c];
define minchange / computed “% change #from min.” center
  style(column) = {cellwidth =  50 pt} style(header) = [just=c];
compute change;
  change=100*( tumsize.sum-bsline)/bsline;
minchange=100*( tumsize.sum- tumsize.min)/tumsize.min;
```
CONCLUSION
The methods in this paper can help us to combine and create various complex tables/listings. This way we may flexibly control the outputs and save some programming time.

REFERENCES

TRADEMARKS
SAS, SAS/BASE, SAS/MACRO software are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

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

Xiaoyu (Sean) Liu
Novartis Oncology
180 Park Ave.
Florham Park, New Jersey 07932 USA
(862)7781099
Xiaoyu.liu@novartis.com