

Strike a Pose! Quick and Easy Camera Ready Reporting with SAS®

Nancy McGarry, Abt Associates Inc.

ABSTRACT

Getting data out to non-programmer staff in a clear non-technical format can present a challenge. Often all the analyst wants to see are numbers presented in a clear and concise manner. Enter **PROC REPORT!**

This ePoster presents a simple SAS program which takes processed data and produces camera ready report results.

INTRODUCTION

This presentation outlines the key ingredients for quickly and painlessly creating tables from an external data set using PROC REPORT. It is a quick crash course in outputting tables that are camera ready.

KNOW THE DATA AND THE TABLE YOU WANT TO CREATE

Before starting to use PROC REPORT create an external data set with the fields that you want reported. Items such as percentages, averages, and standard deviations should be included with a key ID variable. The key ID variable will be used to place the data lines in the correct order. It is also a good idea to have a mock up of what the final table will look like. This helps to position the lines.

CREATE A SERIES OF FORMATS

The formats are used to place the data into the table through the key ID variable. For this program the key ID variable is GENDER.

```
*create formats to position values;
*first which section will this data appear;
*second in each section what line will it appear on;
PROC FORMAT;
value $var2cat
'overall' = '1'
'male'    = '2'
'female'  = '2'
;

value $var2row
'overall' = '1'
'male'    = '1'
'female'  = '2'
;
```

ATTACH THE FORMATS TO THE DATA SET

Simply read in the external data set and apply the positional formats to the key ID variable.

```
*read in data and assign positions;
DATA temp;
  SET sav.sesug2016data;

*assign postions based on an ID key;
*the key for this program is the variable "GENDER";
  rowcat=input (put (gender,$var2cat.),2.);
  rownum=input (put (gender,$var2row.),2.);
run;
```

CREATE THE FRAME WORK OF THE REPORT

These are the rows in the tables. Simply use output statements to create the left most column. One set of statements for each line in the report.

```
*create shell for table - outputting each line that will appear in the table;
*length statement to make sure you can fit stuff;
*first which section will this data appear;
*second in each section what line will it appear on;
DATA tableshell;
  length outcome $ 260; /*make sure this is long enough for formatted
characters*/
  rowcat=1;
  rownum=0;
  boldit=1;
  outcome='This is the Main Heading for the First Section - Overall Numbers';
  output;
  rowcat=1;
  rownum=1;
  boldit=0;
  outcome='      Overall Population in Spreadsheet';
  output;
.....
run;
```

PUT THE DATA FILE AND TABLE SHELL TOGETHER

Merge the 2 files using the key ID variable. Create a final data set with only the variables that will be printed in the report. This file should include any formatting done to create nicer looking output.. This is when changes to the presentation of the data is made: such things as formatting averages to the tenth decimal place; converting percentages from a 0-1 range into a 0-100 range; adding () around a variable.

```

*put the 2 pieces of the puzzle together;
PROC SORT DATA = tableshell;
  BY rowcat rownum;
RUN;
PROC SORT DATA = temp;
  BY rowcat rownum ;
RUN;

DATA temp2 ;
  merge tableshell(in=in1) temp(in=in2);
  BY rowcat rownum ;
RUN;

*get the data to look like we want;
*keep only what should be printed in the report;
DATA temp3(drop =male stdmale female stdfemale age stdage height stdheight gender);
  LENGTH male_mean male_std female_mean female_std age_mean age_std height_mean
  height_std $50;
  set temp2;
  BY rowcat rownum ;

  if age      eq .      then age_mean      = ' '; else age_mean      = (put(age,comma9.1));
  if stdage eq .      then age_std       = ' '; else age_std       =
  ' (' || (put(stdage,comma5.1)) || ') ' ;

```

GENERATE THE REPORT

Using the limited newly formatted data set, add titles and footnotes as needed and use PROC REPORT to generate a table. This version of the program shows how to use a macro variable to populate the title and footnotes. There are a couple of house keeping items that help the program run.

Housekeeping:

```

** some housekeeping;
ODS LISTING CLOSE;

OPTIONS NODATE NONUMBER;

options orientation=landscape leftmargin=.75in rightmargin=.75in nodate nonumber
missing=' ';

ods tagsets.rtf file="&table_name._&sysdate..rtf" path=outfile STYLE=styles.minimal
;

* necessary definitions;
ods escapechar='^';

*put in titels and foot notes;
%let title1=SESUG 2016 Strike A Pose: Camera Ready Strike a Pose! Quick and Easy
          Camera Ready Reporting with SAS®;
%let foot1 = by Nancy McGarry, Abt Associates, 617-349-2361;
%let foot2 = ^{super a} This is redundant;

```

Generating the table:

```

*define STYLE for Table;
*define COLUMNS for Table;
*define DEFINE how to display columns;
*note definition BOLDIT in the compute statements allows bolding;
PROC REPORT DATA=temp3
  STYLE(report)=[rules=none frame= void cellpadding=3pt vjust=b]
  STYLE(header)=[just=center font_face=Arial font_weight=medium font_size=8pt
    vjust=bottom background=white foreground=black frame=hsides]
  STYLE(lines)=[just=left font_face=Arial] SPLIT='|';

/*make sure 1st column non numeric otherwise PROC REPORT outputs 1 line with the
sum of every numeric variable*/
  COLUMNS (' ' outcome) ('Age' age_mean age_std) ('Height' height_mean
height_std) ('Males' male_mean male_std) ('Females' female_mean female_std) boldit
;

  DEFINE boldit / DISPLAY ' ' NOPRINT;

  DEFINE outcome/STYLE(COLUMN)={just=l font_face=Arial vjust=bottom
    font_size=8pt cellwidth=510 }
    STYLE(HEADER)={just=l font_face=Arial font_weight=bold
    font_size=8pt } LEFT;
  DEFINE age_mean / STYLE(COLUMN)={just=r font_face=Arial vjust=bottom
    font_size=8pt cellwidth=105 }

.....
  compute boldit;
    if (boldit eq 1) then call define(_row_,"STYLE","STYLE=[FONT_WEIGHT=BOLD
    BACKGROUND=white borderbottomcolor=white borderrightcolor=white
    borderleftcolor=white bordertopcolor=black]");
  Endcomp;

  title1 font=Helvetica height=10pt bold justify=left "&title1.";
  footnote1 font=Helvetica height=8pt justify=left "&foot1.";
  footnote2 font=Helvetica height=8pt justify=left "&foot2.";
run;

ods tagsets.rtf close;
quit;

```

THIS WILL DELIVER A CAMERA READY TABLE

Due to space constraints not all the code is in the paper, but key examples are. I can provide the program and a csv file with the data that was used by the program.

CONTACT INFORMATION

Nancy McGarry:
 Abt Associates Inc.
 E-mail: nancy_mcgarry@abtassoc.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.