SESUG Paper 179-2018

Table Taming – Using ODS to Simplify Data Presentation

Emily Perry, Rho, Inc.

ABSTRACT

PROC REPORT is a powerful tool, but may be much more than you need. It contains twelve statements and pages of options, making it difficult to display exactly what you want in exactly the way you want it. Consider an alternative method of implementing ODS TRACE and ODS OUTPUT. These SAS® statements can be used to output the results of almost any PROC step into a dataset that can then be manipulated to your heart's content. You have more control over your analysis with a straightforward and intuitive method. In this article, we will discuss using these ODS methods in conjunction with typical PROC steps, such as PROC MEANS and PROC FREQ, along with data manipulation to create tables that you can be proud of.

INTRODUCTION

As SAS programmers, we are constantly knee deep in data. These data are manipulated and analyzed in numerous ways that then must be presented to the investigators in a meaningful way. In clinical research, we present the data and corresponding results using tables, listings, and figures (TLFs). Sometimes it can become tedious to find a simple and straightforward method to present our data in this way. Many people choose to use PROC REPORT to create these kinds of displays, but the learning curve is steep. PROC REPORT contains twelve statements and numerous options that can be overwhelming to even a seasoned SAS user.

An alternative approach is to take advantage of the ODS functionality of SAS. "ODS" stands for "Output Delivery System" and is used in a variety of ways to track and export output within and from the SAS environment. Using ODS TRACE allows you to find the appropriate output to take from your PROC statement. ODS OUTPUT will then translate that result into a dataset that you can manipulate as desired. Although there is less flexibility in the cosmetics, this method gets the desired results presented in an understandable way.

In this paper, we will use a few PROC steps on a test dataset to show the flexibility of using ODS functionality to create meaningful tables.

PROCESS

1. **ODS TRACE**

The first step to find the output needed for your display is to discover the datasets that are available for any given PROC. This can be done using ODS TRACE. When a PROC step is executed, datasets are automatically created in the background. ODS TRACE allows you to see what these datasets are and what is contained in each one. To use ODS TRACE, you simply place "ODS TRACE ON" before your PROC step and "ODS TRACE OFF" after the "run" statement. This will output a list of datasets with their corresponding attributes to the log screen that you can then choose to output (Output 1).

```
Output Added:
-------------
Name:        NObs
Label:       Number of Observations
Template:    Stat.Reg.NObs
Path:        Reg.MODEL1.Fit.SBP_W24.NObs
-------------

Output Added:
-------------
Name:        ANOVA
Label:       Analysis of Variance
Template:    Stat.REG.ANOVA
Path:        Reg.MODEL1.Fit.SBP_W24.ANOVA
-------------

Output Added:
-------------
Name:        FitStatistics
Label:       Fit Statistics
Template:    Stat.REG.FitStatistics
Path:        Reg.MODEL1.Fit.SBP_W24.FitStatistics
-------------

Output Added:
-------------
2

Name:        ParameterEstimates
Label:       Parameter Estimates
Template:    Stat.REG.ParameterEstimates
Path:        Reg.MODEL1.Fit.SBP_W24.ParameterEstimates
-------------
```

Output 1: The Log Output for PROC REG when using ODS TRACE

The portions of the ODS TRACE log output that we are most interested in are “Name” and “Label”. “Name” is the dataset name that you will use within ODS OUTPUT (more on this later). “Label” gives you a description of what the dataset contains. Note that these datasets are not available for manipulation within your SAS environment until you output them to the WORK directory, which we will cover in the next step.

2. ODS OUTPUT

To output your chosen dataset, place the ODS OUTPUT statement within your PROC step. The statement should include the name given in the ODS TRACE log output and the dataset name you want to give it. For example, to acquire the parameter estimates from the PROC REG above, we would add the following code to the PROC step:

```
ods output ParameterEstimates=estimates;
```

This creates the dataset within your working directory.

Most often, you will not want the results in the exact format that SAS gives you. From here, you will want to manipulate the dataset to the format desired to present to your investigator. This is a great time to dust off your basic SAS skills to combine variables, transpose the results, and/or remove unnecessary information from the dataset.

3. MACRO CREATION

This is an optional step but is extremely helpful if you have multiple variables that you want to send through the same PROC step. A macro is a way to run the same set of code with a few different attributes. For example, if you wanted to run PROC MEANS on 10 different numeric variables, you could just change the variable name within a macro call (one line of code to reference all the code included in the macro) for each variable instead of copying and pasting the same code and changing each instance of the variable name. We will not go in depth into this topic, but we will cover a few simple examples in the later section. Please see the reference list if you are interested in learning more.

4. EXPORT

After outputting and manipulating all desired results, you will want to export the table out of the SAS environment. Again, ODS functionality can be used for this purpose. To export to a rich text format (rtf) file, ODS RTF is used. To export to a portable document format (pdf), ODS PDF is used. To view more output options, see "SAS 9.4 Output Delivery System: User's Guide, Fifth Edition". Similar to ODS TRACE, these statements must be opened and closed with the desired output between them. When opening, the file path for the output location must be specified as follows:

```
ods rtf file="S:\Conferences\SESUG 2018\Table1.rtf";
 title "Table 1";
 proc print data=data;
 run;
ods rtf close;
```

EXAMPLE

TEST DATASET

To demonstrate this process and the flexibility of the ODS functionality, we will create a test dataset to manipulate. This dataset contains a fabricated study for a drug to lower blood pressure, specifically systolic blood pressure (SBP). Table 1 gives the dataset specifications so that you can familiarize yourself with the variables we will be using. This dataset contains both numeric and categorical variables that we would potentially be interested in presenting in a meaningful way to our investigator.

Variable	Description	Type
USUBJID	Unique Subject Identifier	Char
TRT	Assigned Treatment (Drug, Placebo)	Char
SEX	Gender (F, M)	Char
AGE	Age	Num
SBP_W1	Systolic Blood Pressure (SBP) at Week 1	Num
SBP_W24	Systolic Blood Pressure (SBP) at Week 24	Num
CH_SBP	Change in SBP from Week 1 to Week 24	Num
NORMFL	SBP in Normal Range at Week 24? (Y, N)	Char

CLSIGFL	Was Change Clinically Significant? (>= 30 mmHg drop in SBP) (Y, N)	Char
LOWFL	Final SBP Low? (Y, N)	Char

Table 1: Dataset Specifications for the Test Dataset

To recreate this dataset, use the code found in the Appendix.

In this simulated study, we are interested in seeing how SBP changes over 24 weeks. We want to present the summary statistics for variables of interest. Continuous variable summaries will display means with standard deviations, and categorical variable summaries will display counts and percentages (Table 2).

Variable	Drug	Placebo
Systolic Blood Pressure (SBP) - Week 1	XX.X (X.XX)	XX.X (X.XX)
Systolic Blood Pressure (SBP) - Week 24	XX.X (X.XX)	XX.X (X.XX)
Change in SBP - Week 1 to Week 24	XX.X (X.XX)	XX.X (X.XX)
Final SBP in Normal Range?		
N	X (XX.X)	X (XX.X)
Y	X (XX.X)	X (XX.X)
Change in SBP Clinically Significant?		
N	X (XX.X)	X (XX.X)
Y	X (XX.X)	X (XX.X)
SBP Low at Week 24?		
N	X (XX.X)	X (XX.X)
Y	X (XX.X)	X (XX.X)

Table 2: Mock Table Output for Test Dataset

For the continuous variables, we will implement PROC MEANS to obtain the mean and standard deviations (std). For the categorical variables, we will implement PROC FREQ to obtain counts and percentages. In both of these processes, we will implement the steps previously discussed: 1) ODS TRACE, 2) ODS OUTPUT, 3) MACRO creation, and 4) Exportation.

PROC MEANS

For this example, we are first interested in the mean and standard deviation (std) results for SBP at Week 1, SBP_W1. Using ODS TRACE, we discover the output options from the PROC MEANS step:

```
ods trace on;
proc means data=TEST mean std;
   Class TRT;
   Var SBP_W1;
run;
ods trace off;
```

This gives one output option called “Summary”. We will then use this in our output statement to create a dataset called “means”:

```
proc means data=TEST mean std;
   Class TRT;
   Var SBP_W1;
   ods output Summary=means;
run;
```

This gives a dataset that contains the mean and std for both treatment groups. For our table, we want these results presented as “mean (std)” and each treatment group to be a column. Thus, we manipulate as follows:

```
data means2;
   set means;
   SBP_W1=strip(put(SBP_W1_MEAN,8.1))||
   "("||strip(put(SBP_W1_STDDEV,8.2))||")";
run;

proc transpose data=means2 out=SBP_W1;
   var SBP_W1;
   id trt;
run;
```

This gives us the statistics needed for the first row of our table (Output 2).

	NAME	DRUG	PLACEBO
1	SBP_W1	140.3 (8.45)	140.3 (5.95)

Output 2: Dataset for SBP_W1 after Data Manipulation

Although this is a great start, we have two other continuous variables, SBP_W24 and CH_SBP, which we need to obtain the same information for. This is where a macro becomes very useful. All of the code we previously submitted for SBP_W1 can be placed within a macro. All references to “SBP_W1” will be replaced with a macro variable call “&var”. The first line of the macro sets the macro name as “means” and defines the macro variable to be “var”, where you specify the variable to be used for analysis.

```
%macro means(var);
      proc means data=TEST mean std;
        class TRT;
        var &var;
        ods output Summary=&var;
      run;

      data &var;
        set &var;
        &var=strip(put(&var._MEAN,8.1))||
        "("||strip(put(&var._STDDEV,8.2))||")";
      run;

      proc transpose data=&var out=&var;
        var &var;
        id trt;
```

```
    run;
%mend means;

%means(SBP_W1)
%means(SBP_W24)
%means(CH_SBP)
```

The macro “means” creates a dataset by the same name that includes the summary statistics needed for the table. Running the macro call, i.e. %means(), creates each output. We will come back to these results to combine them with the results from PROC FREQ.

PROC FREQ

Next, we want to create counts and percentages for all of our categorical variables. We will start with the amount of people with and without SBP in the normal range at Week 24, NORMFL. Using ODS TRACE once again, we discover the output options from the PROC FREQ step:

```
ods trace on;
  proc freq data=TEST;
    table NORMFL*TRT;
  run;
ods trace off;
```

This gives one output option called “CrossTabFreqs”. We will then use this in our output statement to create a dataset called “freq”:

```
proc freq data=TEST;
  table NORMFL*TRT;
  ods output CrossTabFreqs=freq;
run;
```

This dataset “freq” contains the frequency, percentages, and missing values for both treatment groups. For our table, we want these results presented as “count (%)”. So, we only want the frequencies and the column percentages, and each treatment group to be a column. Also, we want an empty row containing only the variable name. Thus, we manipulate as follows:

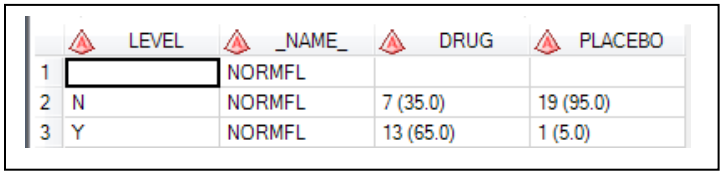
```
proc sort data=freq2;
  by normfl trt;
run;
proc transpose data=freq2 out=normfl;
  var normfl_;
  id trt;
  by normfl;
run;

data normfl;
  Var="NORMFL";
  output;
  set normfl;
run;

data normfl(keep=_NAME_ LEVEL DRUG PLACEBO);
  length LEVEL $20;
  set normfl;
  _NAME_=VAR;
  LEVEL=NORMFL;
```

```
run;
```

This gives us the statistics needed for the first row of our table (Output 3).



	LEVEL	_NAME_	DRUG	PLACEBO
1		NORMFL		
2	N	NORMFL	7 (35.0)	19 (95.0)
3	Y	NORMFL	13 (65.0)	1 (5.0)

Output 3: Dataset for NORMFL after Data Manipulation

Again, we will want to execute a macro to create the other categorical variables, CLSIGFL and LOWFL. This time all references to “NORMFL” will be replaced with the macro variable call “&var”:

```
%macro frq(var);
   proc freq data=TEST;
      table TRT*&var;
      ods output CrossTabFreqs=freq;
   run;
                              ;
   data freq2;
      set freq(where=(^missing(&var) and ^missing(TRT)));
      &var._=strip(put(FREQUENCY,8.0))||
      "("||strip(put(COLPERCENT,8.1))||")";
   run;
   proc sort data=freq2;
      by &var trt;
   run;
   proc transpose data=freq2 out=&var;
      var &var._;
      id trt;
      by &var;
   run;

   data &var;
      Var="&var";
      output;
      set &var;
   run;

   data &var(keep=_NAME_ LEVEL DRUG PLACEBO);
      length LEVEL $20;
      set &var;
      _NAME_=VAR;
      LEVEL=&var;
   run;
%mend frq;

%frq(NORMFL)
%frq(CLSIGFL)
%frq(LOWFL)
```

Now we have the results needed from the three categorical variables that we can combine with our PROC MEANS results from the last section.

EXPORT

Once all of the results have been created, they need to be combined before exporting. We also want to make sure that the text says exactly what we specified in the mock. We will do the desired remaining manipulation and then export using ODS RTF as follows:

```
data final(drop=_NAME_ LEVEL);
  length Variable $200;
  set SBP_W1 SBP_W24 CH_SBP NORMFL CLSIGFL LOWFL;
  if _NAME_="SBP_W1" then Variable="Systolic Blood Pressure (SBP) -
  Week 1";
  if _NAME_="SBP_W24" then Variable="Systolic Blood Pressure (SBP)
  - Week 24";
  if _NAME_="CH_SBP" then Variable="Change in SBP - Week 1 to Week
  24";
  if _NAME_="NORMFL" then Variable="Final SBP in Normal Range?";
  if _NAME_="CLSIGFL" then Variable="SBP Change Clinically
  Significant?";
  if _NAME_="LOWFL" then Variable="Final SBP Low?";
  if ^missing(LEVEL) then Variable="  "||LEVEL;
run;

ods rtf file="H:\Conferences\SESUG\SESUG 2018\Table.rtf";
title "Table 1: Results by Treatment Group";
  proc print data=final noobs;
  run;
ods rtf close;
```

Table 1: Results by Treatment Group

Variable	DRUG	PLACEBO
Systolic Blood Pressure (SBP) - Week 1	140.3 (8.45)	140.3 (5.95)
Systolic Blood Pressure (SBP) - Week 24	117.1 (13.24)	134.9 (7.04)
Change in SBP - Week 1 to Week 24	-23.2 (14.73)	-5.4 (4.84)
Final SBP in Normal Range?		
N	7 (35.0)	19 (95.0)
Y	13 (65.0)	1 (5.0)
SBP Change Clinically Significant?		
N	13 (39.4)	20 (60.6)
Y	7 (100.0)	0 (0.0)
Final SBP Low?		
N	19 (48.7)	20 (51.3)
Y	1 (100.0)	0 (0.0)

Output 4: Final Table Created

The resulting table is then placed and named as specified in the file path given in ODS RTF. The output produced matches what was in our original mock (Output 4). We are finished!

CONCLUSION

In this paper, we explored how to use ODS TRACE and ODS OUTPUT to create a simple summary table. The fun does not end there! The greatest benefit to using this method is that it is flexible and can be used on almost any PROC step to get the results you desire. This method is also simple and easy to implement. More complex PROC implementation will have multiple options to choose from when using ODS TRACE to get the results you are interested in summarizing.

However, there are limitations to this method, particularly in the export options. There is not a substantial amount of cosmetic flexibility in the output. When exporting, you will get a simple table in a format similar to what is shown in the SAS environment. It is also important to note that the name of the output created will change based on the options you specify within the PROC, so always check ODS TRACE after altering the PROC step.

If you are not satisfied with the output that this method gives you, go explore PROC REPORT where you can make the output fit the particular style you are looking for. ODS TRACE can be using in conjunction with PROC REPORT as well to get your desired output. Whatever method you choose, simply using ODS TRACE with ODS OUTPUT will never steer you wrong.

REFERENCES

SAS Institute Inc. 2017. Base SAS® 9.4 Procedures Guide, Seventh Edition. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2018. SAS® 9.4 Macro Language: Reference, Fifth Edition. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2018. SAS® 9.4 Output Delivery System: User's Guide, Fifth Edition, Fifth Edition. Cary, NC: SAS Institute Inc.

Slaughter, Susan J. and Lora D. Delwiche. 2004. "SAS® Macro Programming for Beginners" Proceedings of the Thirtieth Annual SAS® Users Group International, 243-29. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Emily Perry
Rho, Inc.
emily_perry@rhoworld.com

APPENDIX

TEST DATASET CODE

```
proc format;
      value $YN "Y"="Yes"
                "N"="No";
run;

data TEST;
      input USUBJID $ TRT $ SEX $ AGE SBP_W1 SBP_W24;
      CH_SBP=SBP_W24 - SBP_W1;
      if 90 <= SBP_W24 <= 120 then NORMFL = "Y";
      else NORMFL="N";
      if CH_SBP<=-30 then CLSIGFL="Y";
      else CLSIGFL="N";
      if SBP_W24 < 90 then LOWFL = "Y";
      else LOWFL="N";
      format NORMFL $YN. CLSIGFL $YN. LOWFL $YN.;
      datalines;
    A01 DRUG M 27 140 123
    A02 DRUG F 34 134 105
    A03 DRUG F 22 138 88
    A04 DRUG M 45 145 110
    A05 DRUG M 38 135 132
    A06 DRUG F 34 133 115
    A07 DRUG M 40 140 138
    A08 DRUG M 42 130 115
    A09 DRUG F 34 143 118
    A10 DRUG M 28 155 111
    A11 DRUG M 40 135 123
    A12 DRUG F 25 134 115
    A13 DRUG F 37 134 92
    A14 DRUG M 71 145 115
    A15 DRUG M 39 130 137
    A16 DRUG F 50 138 115
    A17 DRUG M 45 160 138
    A18 DRUG M 36 135 120
    A19 DRUG F 27 148 112
    A20 DRUG M 54 153 119
    B01 PLACEBO M 27 142 141
    B02 PLACEBO F 34 132 118
    B03 PLACEBO F 22 134 128
    B04 PLACEBO M 45 140 135
    B05 PLACEBO M 38 138 132
    B06 PLACEBO F 34 143 130
    B07 PLACEBO M 40 150 140
    B08 PLACEBO F 42 135 135
    B09 PLACEBO F 34 140 136
    B10 PLACEBO M 28 142 140
    B11 PLACEBO M 50 147 140
    B12 PLACEBO F 34 135 123
```

```
    B13 PLACEBO F 25 130 128
    B14 PLACEBO M 51 145 146
    B15 PLACEBO M 40 139 132
    B16 PLACEBO F 29 140 134
    B17 PLACEBO M 38 154 143
    B18 PLACEBO F 39 135 137
    B19 PLACEBO F 43 142 136
    B20 PLACEBO M 33 142 143
    ;
run;
```