Paper 075-2011

Writing SAS® Programs with SAS and Excel

Laiju Zhang, MDCI, North Attleboro, MA, USA

Abstract

This paper presents SASWriter the macro system for writing SAS programs that produce tables, listings, and figures for standard FDA submission. It explains the structure of the macro system by identifying the functionality of each part. It also discusses the consequences the SAS macro has for SAS programming in general in terms to programmers' relation to other team members.

Key words: macro, SAP, programmers, code generation.

1 Introduction

Believe it or not, long painful debugging hours or days might just lead to the discovery of two types of seemingly naïve errors: typos, and misplacements. You type one letter wrong in your datasets name or variable name, your SAS programs will crash. You place your dataset in the wrong place vs. where it is supposed to be, the smart SAS will never find it even it is in the closest neighbors.

These two lower levels mistakes popped up in many of projects so often that I decided to reduce them – if not to eliminate them. I therefore set up my goal of letting SAS take care of the exact names it requires for each program, and places the program and dataset exactly where they should be. In the process of achieving this goal, I built a macro system that generates SAS programs based on the datasets information and analysis decision that is stored in excel.

This paper is a presentation of the macro system which consists of the following parts:

2 Structure of SASWriter

The structure of this macro can be summarized in the following table:

Table 1: SASWriter

No.	Name of Macro	Input	Output
1	%pop	subject info	group size, filtered subject group
2	%conti	continuous variable	descriptive statistics

3	%cate	categorical variable	descriptive statistics
4	%pop		
5	%report	final dataset	pdf, rft, xml
6	%ae	ae dataset	pdf, rft, xml
7	%lab	lab dataset	pdf, rft, xml

The macros listed in the above table are lower level macros and are designed to handle only one variable each time for each dataset. The outputs they produce are standard formats. For continuous variable, they include n, mean, std, max, min, etc that are output by **proc univariate**. For categorical variables, they are number and percents for each variables produced by **proc freq**. Here are more detailed explanation for each item:

1) %pop takes in a dataset summarizing each subject's information which includes but not limited subject unique id, demographic information, population information, and base line information. By passing different values to each parameter, %pop controls what population to be used as the base for analysis. At the same time, it produces the sample size for each group and stores the information in macro variables to be used by %conti, %cate, and %report in their calculations and results presentations.

2) %conti handles continuous variables. It takes an analysis–ready dataset as and other parameters as input, and outputs report-ready dataset to be combined with other outputs to produce a final dataset that is to be used by %report.

3) %cate behaves similarly as %conti except that it deals with categorical variables, not continuous variables.

4) %pvalue macro is called by either %conti or %cate to produce the p values according to the specified statistical models from statistical analysis plan. The results are inserted into the output dataset that is passed to %conti or %cate.

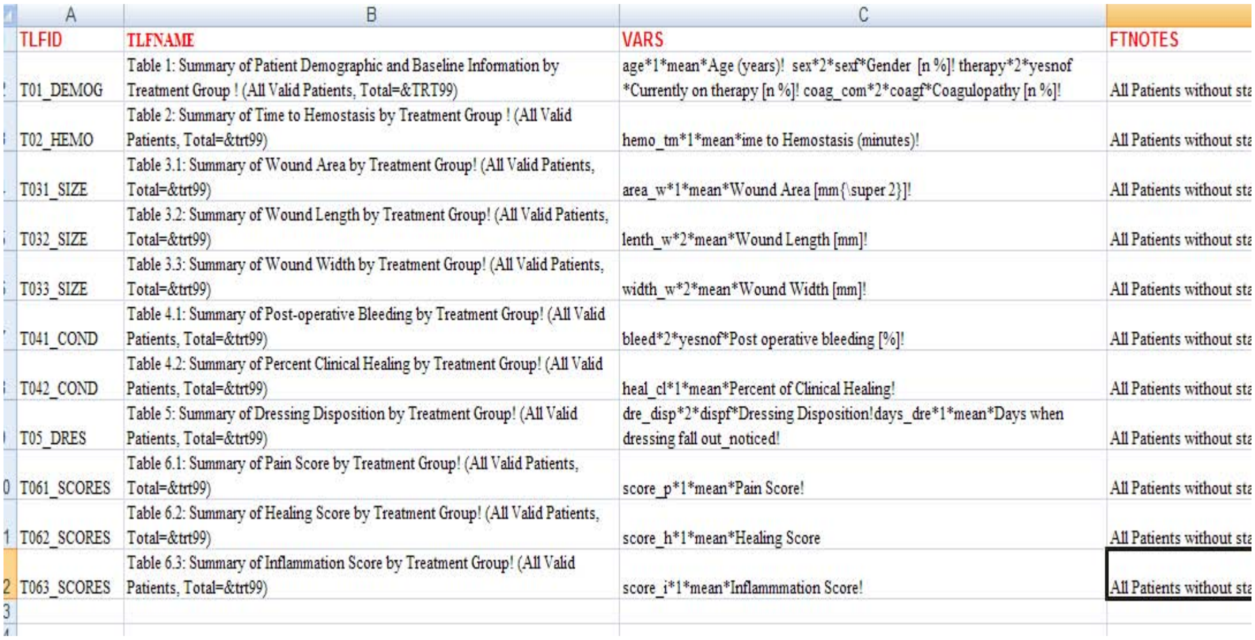
5) %report is based on proc report. The input dataset is the dataset that combines all the %conti and %cate outputs. The titles and foot notes are left as parameters that are to be filled at run time.

6) %ae macro is constructed to handle different types of AE tables, based on %conti, %cate, %pvalue, and %report.

7) %lab is used to deal with different formats and layouts for lab tables. Like %ae, it is also built on top of %conti, %cate, %pvalue, and %report.

2.1 *Specifications for Tables, Listings, and Figures*

Figure 1 TLF Specifications



A	B	C	
TLFID	TLFNAME	VARS	FTNOTES
T01_DEMOG	Table 1: Summary of Patient Demographic and Baseline Information by Treatment Group ! (All Valid Patients, Total=&TRT99)	age*1*mean*Age (years)! sex*2*sexf*Gender [n %]! therapy*2*yesnof *Currently on therapy [n %]! coag_com*2*coagf*Coagulopathy [n %]!	All Patients without sta
T02_HEMO	Table 2: Summary of Time to Hemostasis by Treatment Group ! (All Valid Patients, Total=&trt99)	hemo_tm*1*mean*ime to Hemostasis (minutes)!	All Patients without sta
T031_SIZE	Table 3.1: Summary of Wound Area by Treatment Group! (All Valid Patients, Total=&trt99)	area_w*1*mean*Wound Area [mm{\super 2}]!	All Patients without sta
T032_SIZE	Table 3.2: Summary of Wound Length by Treatment Group! (All Valid Patients, Total=&trt99)	lenth_w*2*mean*Wound Length [mm]!	All Patients without sta
T033_SIZE	Table 3.3: Summary of Wound Width by Treatment Group! (All Valid Patients, Total=&trt99)	width_w*2*mean*Wound Width [mm]!	All Patients without sta
T041_COND	Table 4.1: Summary of Post-operative Bleeding by Treatment Group! (All Valid Patients, Total=&trt99)	bleed*2*yesnof*Post operative bleeding [%]!	All Patients without sta
T042_COND	Table 4.2: Summary of Percent Clinical Healing by Treatment Group! (All Valid Patients, Total=&trt99)	heal_cl*1*mean*Percent of Clinical Healing!	All Patients without sta
T05_DRES	Table 5: Summary of Dressing Disposition by Treatment Group! (All Valid Patients, Total=&trt99)	dre_disp*2*dispf*Dressing Disposition!days_dre*1*mean*Days when dressing fall out_noticed!	All Patients without sta
T061_SCORES	Table 6.1: Summary of Pain Score by Treatment Group! (All Valid Patients, Total=&trt99)	score_p*1*mean*Pain Score!	All Patients without sta
T062_SCORES	Table 6.2: Summary of Healing Score by Treatment Group! (All Valid Patients, Total=&trt99)	score_h*1*mean*Healing Score	All Patients without sta
T063_SCORES	Table 6.3: Summary of Inflammation Score by Treatment Group! (All Valid Patients, Total=&trt99)	score_i*1*mean*Inflammmation Score!	All Patients without sta

As shown in Figure 1, table specification includes all variables to be used and statistic methods to be applied. They follow exactly what SAP (Statistical Analysis Plan) specifies. These specifications are stored in a simple excel file that can be read and modified by non SAS programmers.

The spread sheet serves two purposes: 1) to provide information as input to produce SAS programs when the project has just started; 2) to update the titles, footnotes during the process of the study when updating of the titles and footnotes are necessary.

1) Proc contents to the analysis datasets and bring them into xls formats.

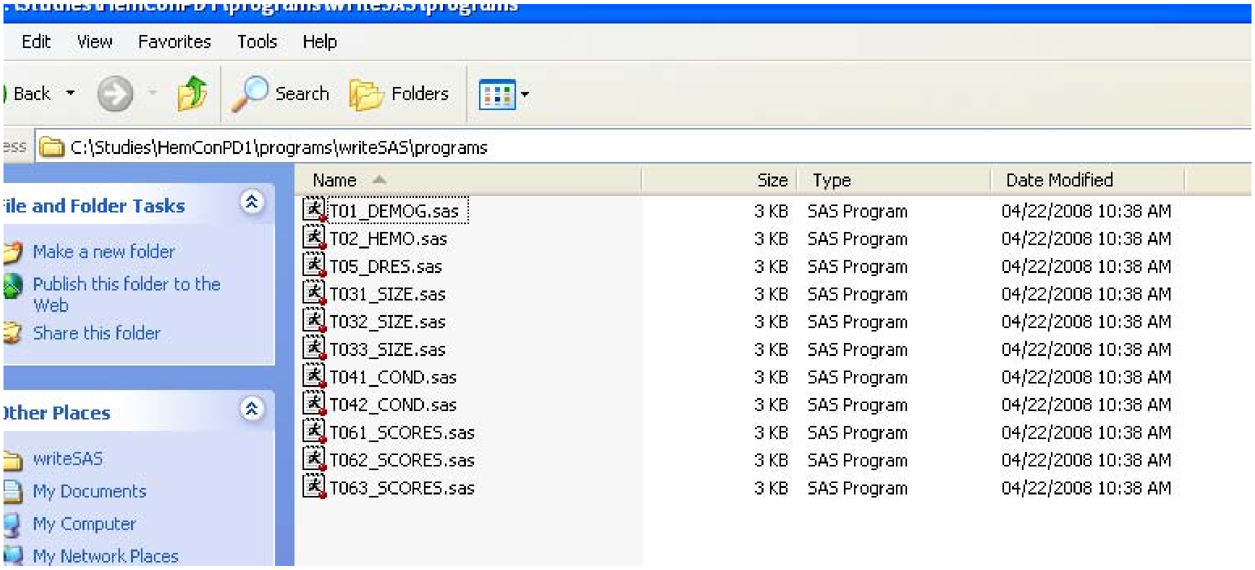
2) Small excel macros to populate datasets names and variables to be clicked to go into each spread spec file.

3) Identify each table with variable, label, and stat methods for producing p values.

2.2 Part III: %SASWriter macro

%SASWriter macro is the commander in chief of the whole macro system. It read s in the excel file that contains all the table specifications, it then produces a standard SAS program that mainly consists of the lower level macros: %cate , %conti, analysis programs for each table/listing, and figures, as shown in the Figures 2, 3.

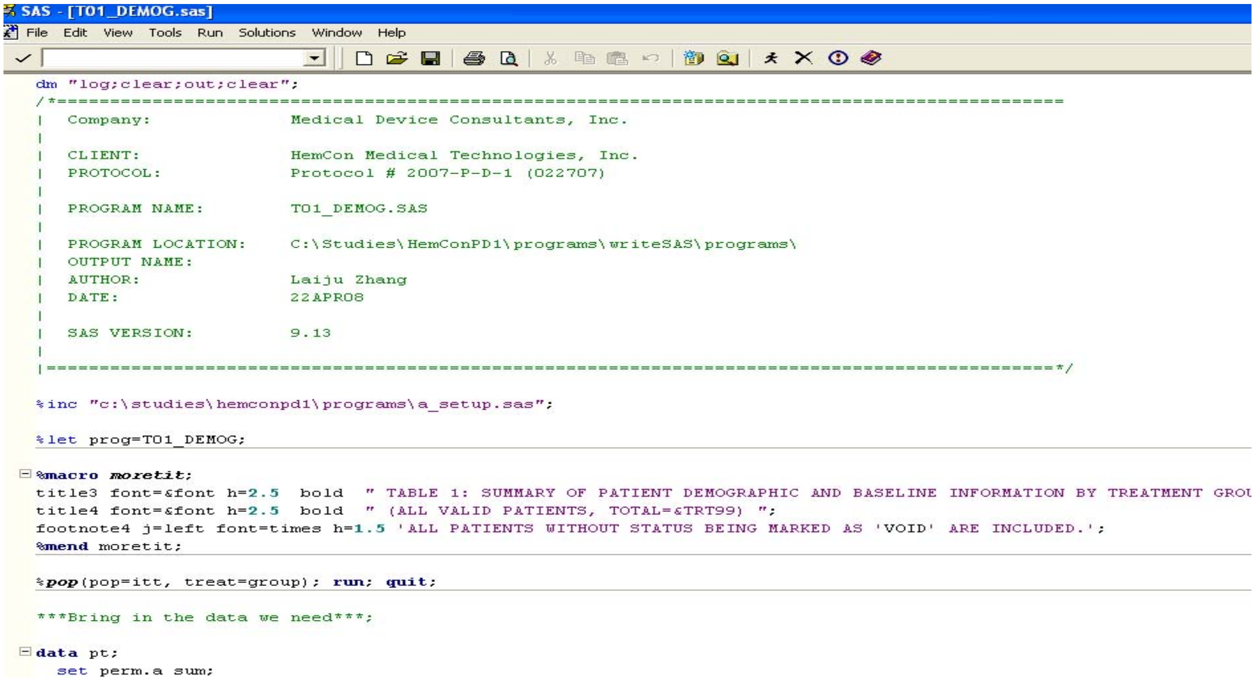
Figure 2. List of SAS files produced by %SASWriter



ess C:\Studies\HemConPD1\programs\writeSAS\programs

Name	Size	Type	Date Modified
T01_DEMOG.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T02_HEMO.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T05_DRES.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T031_SIZE.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T032_SIZE.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T033_SIZE.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T041_COND.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T042_COND.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T061_SCORES.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T062_SCORES.sas	3 KB	SAS Program	04/22/2008 10:38 AM
T063_SCORES.sas	3 KB	SAS Program	04/22/2008 10:38 AM

Figure 3 Individual SAS program that is produced by %SASWriter.



```
dm "log;clear;out;clear";
/*=====================================================================================
|   Company:                 Medical Device Consultants, Inc.
|
|   CLIENT:                  HemCon Medical Technologies, Inc.
|   PROTOCOL:                Protocol # 2007-P-D-1 (022707)
|
|   PROGRAM NAME:            T01_DEMOG.SAS
|
|   PROGRAM LOCATION:        C:\Studies\HemConPD1\programs\writeSAS\programs\
|   OUTPUT NAME:
|   AUTHOR:                  Laiju Zhang
|   DATE:                    22APR08
|
|   SAS VERSION:             9.13
|
|=====================================================================================*/

%inc "c:\studies\hemconpd1\programs\a_setup.sas";

%let prog=T01_DEMOG;

%macro moretit;
title3 font=&font h=2.5  bold  " TABLE 1: SUMMARY OF PATIENT DEMOGRAPHIC AND BASELINE INFORMATION BY TREATMENT GROU
title4 font=&font h=2.5  bold  " (ALL VALID PATIENTS, TOTAL=&TRT99) ";
footnote4 j=left font=times h=1.5 'ALL PATIENTS WITHOUT STATUS BEING MARKED AS 'VOID' ARE INCLUDED.';
%mend moretit;

%pop(pop=itt, treat=group); run; quit;

***Bring in the data we need***;

data pt;
  set perm.a sum;
```

In addition, the macro copies the following files to the study directory as utility tools:

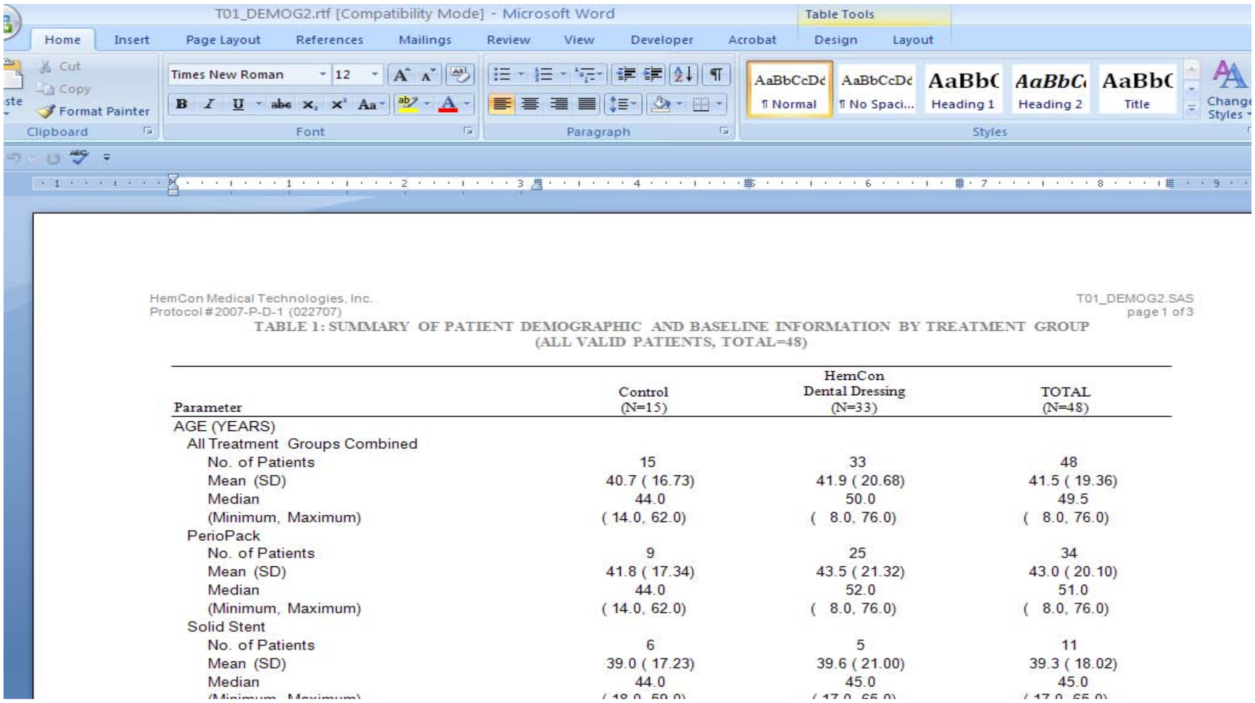
1) check.pl is a script written in perl that is used to find “errors”, “ warnings” or other suspicious information from SAS log files and writes a report on these findings.

2) TOC.SAS is a SAS program that produces table of contents for all the listings, graphs, and tables that have been generated.

3) Run.bat file that is a dos bat file that will run all SAS programs in the project directory.

After the SAS programs are in place, we can produce various types of files. One example is illustrated by Figure 4.

Figure 4 Example of Table in RTF format



HemCon Medical Technologies, Inc.
Protocol # 2007-P-D-1 (022707)

T01_DEMOG2.SAS
page 1 of 3

TABLE 1: SUMMARY OF PATIENT DEMOGRAPHIC AND BASELINE INFORMATION BY TREATMENT GROUP
(ALL VALID PATIENTS, TOTAL=48)

Parameter	Control (N=15)	HemCon Dental Dressing (N=33)	TOTAL (N=48)
AGE (YEARS)			
All Treatment Groups Combined			
No. of Patients	15	33	48
Mean (SD)	40.7 (16.73)	41.9 (20.68)	41.5 (19.36)
Median	44.0	50.0	49.5
(Minimum, Maximum)	(14.0, 62.0)	(8.0, 76.0)	(8.0, 76.0)
PerioPack			
No. of Patients	9	25	34
Mean (SD)	41.8 (17.34)	43.5 (21.32)	43.0 (20.10)
Median	44.0	52.0	51.0
(Minimum, Maximum)	(14.0, 62.0)	(8.0, 76.0)	(8.0, 76.0)
Solid Stent			
No. of Patients	6	5	11
Mean (SD)	39.0 (17.23)	39.6 (21.00)	39.3 (18.02)
Median	44.0	45.0	45.0
(Minimum, Maximum)	(18.0, 59.0)	(17.0, 65.0)	(17.0, 65.0)

3 Discussions

In retrospect, designing SASWriter follows the code generation approach in software development world. It is similar to traditional SAS macro approach in the following ways:

- Let programmers concentrate on concepts, rather than coding, standard coding is moved to various kinds of macros.

- Programmatically retrieve data information in the construction of SAS programs to avoid typos.

- Programmatically place programs, specify paths info so that SAS will not get lost as often as it happens in the manual programming case.

However, SASWriter deviates from the macro approach in the following aspects:

- Participation of TFF generation by other team members. The Excel that specifies each TLF can be updated by SAS programmers and non SAS programmers. Other team members (medical writers, statisticians, etc.) can update the titles, footnotes, etc. Or these footnotes can be retrieved from SAP programmatically. As a result, SAS programmers can concentrate on programming logics (which are our strength), not texts (which are our weakness).

- Updating of SAS programs. In contrast to the traditional macro approach which produces the final TLFs, SASWriter take additional step: it only produce SAS programs. As a result, these programs can be updated, QCed by colleagues, and can be sent to FDA for submission purpose. Thus, the logics that are applied in each program are more transparent and easier to be updated.

- Different from overall macros where all logics are placed in one macro, this approach only partially standardizes macros utilities to the level of individual variables, each table and listing are still produced by each SAS program.

4 Contact Information

Laiju Zhang, PhD

Sr. Manager of Biostatistics & SAS programming/Sr Biostatistician
Medical Device Consultants, Inc.

49 Plain st, North Attleboro, MA 02760
(508) 316 7163

(508) 643-2237 Fax
LZhang@mdci.com
http://www.mdci.com

5 References

To download the compiled macros that are discussed in this paper, go to visit the front page of www.SASLab.org for the source code for this paper (http://www.saslab.org/?q=node/60).

.