Paper 216 -28

"From Data to Analysis, Results and Reports" A Researcher's Dilemma but A Programmer's challenge.

Shabnam Mehra, University of South Florida, Tampa, FL

ABSTRACT

Let SQL, ODS and Macros make it easy for you. Automation is the keep to save time and repetitive work for a programmer. Administrative data are always a challenge to clean, format and use effectively in analysis. Using macros to clean, format and combine administrative data saves time and energy for a programmer. Proc SQL® within a macro can help to combine multiple tables, calculate summary variables and generate more meaningful queries and tables. Using SQL to generate tables can help you create more "user-friendly" and "analysis-friendly" datasets. Proc Freq® and Proc Means® used to generate descriptive and basic statistics can be automated within a macro. The output objects or tables from the analysis can be sent to one or more destinations, such as html, rtf (word) or csv (excel) using ODS statements. Automation using macros saves repetitive coding and ODS helps to generate high quality reports in short time.

INTRODUCTION

At our data center Data managers often receive several secondary data sets from multiple agencies and have different layouts and formats. These datasets are used in research at the institute to answer questions pertaining to mental health services usuage. Secondary data analysis is an important part of health care field. It helps to answer reaseach questions without going through the hassle of primary data collection. However, analysis from secondary date is very complicated and data needs extensive cleaning. A programmers challenge is to clean, format and combine these administrative data. Every year we receive requests to produce annual service use reports and cost associated with these services. Very often I have to cross different state systems to study the penetration among the different systems. There are several reasons to automate this process, namely SAS analytical results will be automatically inserted into the pre-defined Excel reports. Firstly, it is possible to create these reports under a tight budget as soon as possible. Secondly, there may be frequent changes in the report format by client over the years. An analyst and client work closely to develop plans to create these reports. A programmer writes SAS Macros to clean, format and combine administrative datasets. We use Proc SQL® to combine multiple tables, calculate summary variables and generate queries and tables. Automated Proc Freq® and Proc Means® are used to calculate predefined statistics. Finally, the progammer uses ODS statements to output tables to Excel and word destinations. The code once developed has helped the datacenter to produce reports more efficiently in less time.

A PROGRAMMER'S CHALLENGE

It is a researcher's dilemma to produce high quality reports using multiple datasets to answer research questions in short time. However, it is a programmer's challenge to accomplish this task efficiently in crunch time without repetative coding and under a tight budget. The code described here is performed on secondary data containing cost associated with services. The code uses exam dates in dataset one to determine service utilization within the dataset two six months before and after an exam. We are specifically interested in utilities such as demographics of the people in the two systems and cost associated with services six months prior and post to the exam. The analyst must have in depth knowledge of the data layout and the purpose for which the data was primarily collected. Often the data layout must be manipulated to accommodate the research objectives. For instance, each claim in the datasets' original form may represent a billing period rather than a treatment period, which would be more useful for research purposes. Close management of variables such as dates, cost, and services is mandatory when using secondary data to answer research questions. The techniques in this paper apply to health care agencies and pharmaceutical industry but may easily be applied to other disciplines.

MACROS TO OUTPUT CLEAN DATA

Data cleaning is a necessary evil and is the real challenge for a programmer. Using macros to clean, format and combine administrative data saves time and energy for a programmer. At the data center we receive data from different agencies and perform secondary data analysis. The intent of secondary analysis is to extract pertinent mental health policy directives from data that was collected for other objectives. For example, mental health claims data that was collected for accounting purposes can be used judiciously to answer mental health policy research questions. SAS Macro are a useful tool to manipulate the data layouts and come in very handy to do this in short span of time. Macros are developed at data center to clean and manipulate the layout, put standard formats and combine different datasets. Macros help to eliminate repetative coding and hence, save time of a programmer. Macros be used to put standard formats across different datasets and hence eliminate the dilemma of different formats across different data systems. Mutliple macros or single macro can be used to accomplish this task.

PROC SQL® TO GENERATE "USER-FRIENDLY" TABLES

SQL in SAS has made life of programmers so much easier. Proc SQL® has many uses, for this paper it has been very helpful in combining multiple tables to one table, calculate summary variables and generate queries or SAS tables. Proc SQL® helps a programmer to eliminate repetitive and extensive coding and also is faster than general data set code. Using SQL to generate tables can help you create more "user-friendly" and "analysis-friendly" datasets.

AUTOMATED PROC FREQ® AND PROC MEANS®

Any predefined statistics can be automated using a Macro and to save time and repetitive programming. Descriptive and basic statistics are generated at the data center with help of Proc Freq® and Proc Means® and are automated within a macro.

ODS STATEMENTS TO OUTPUT OBJECTS OR TABLES

The output objects or tables from the analysis can be sent to one or more destinations, such as html, rtf (word) or csv (excel) using ODS statements. ODS helps to generate high quality reports in short time. At the data center we very frequently use ODS to generate tables in excel for our clients and word tables to be inserted in reports and manuscript of papers.

THE CODING

The challenge is to attain a record from the two differently formated datasets coming from different state agencies. Once this challenge is completed, descriptive and basic statististics can easily be calculated. Macros and SQL statements can be easily employed to automate the process. The first step in the code is to obtain a SAS datasets from access database using Proc Import. Next in the Macro " D_FORMATS" the variables of input datasets are formatted to standard formats so that the input datasets can be combined.

Shown here is the formats of the original datasest as it obtained from the two agencies:

Data1

Variable	Format
ID	12.
Csex	8.
Crace	8.
Locexam	8.
Certype	8.

Data2

Variable	Format
ID	$9.
sex	$1.
race	$1.
D_cnty	$2.
State	$2.

The format coding process will create the following data1:

Data1

Variable	Format
ID	$9.
sex	$1.
race	$1.
PLocexam	$1.
Ctype	$1.

The next step is to combine the two datasets, using the unique recipient identification variable (ID) or SSN. The data are merged using IN statement. Then Proc SQL® is used to determine service utilization within the dataset two six months before and after an exam originally in data1. At this point two separate utilization datasets are created, one with six months utilization prior to exam date and one with six months utilization post exam date. The code is in Macro "Merge_it", and the Macro helps to eliminate repetative coding and changes to the datasteps. The macro can be submitted to combine any two datasets with common formats and produce results in short time.

Data1

ID*	Exm_Date
123	7/3/2001
320	7/15/2001
456	8/24/2001
515	7/15/2001
620	7/15/2001
780	8/24/2001

Data2

ID*	Serv_date
123	5/8/2001
123	9/8/2001
320	3/22/2001
320	8/30/2001
320	2/24/2002
456	5/14/2001
456	11/14/2001
515	2/24/2002
618	2/8/2002

**ID's used here are SSN only last three digits are printed on the table*

"Combo":

ID*	Exm_Date	Serv_date
123	7/03/01	5/08/01
123	7/03/01	9/08/01
320	7/15/01	3/22/01
320	7/15/01	8/30/01
320	7/15/01	2/24/02
456	8/24/01	5/14/01
456	8/24/01	11/14/01
515	7/15/01	2/24/02

"Combo_Bef":

ID*	Exm_Date	Beg_Date	Serv_date
123	7/03/01	1/03/01	5/08/01
320	7/15/01	1/15/01	3/22/01
456	8/24/01	2/24/01	5/14/01

"Combo_aft"

ID*	Exm_Date	Aft_Date	Serv_date
123	7/03/01	1/03/02	9/08/01
320	7/15/01	1/15/02	8/30/01
456	8/24/01	2/24/02	11/14/01

**ID's used here are SSN only last three digits are printed on the table*

Once these two datasets with six months prior and post exam datasets are obtained tables are generated with descriptive statistics. Proc SQL® is used to first create tables "Demo_Freq_bef" and "Demo_Freq_aft" into which the frequencies are inserted by the Macro "freq_it". The macro "analyse_it" is used to generate means, standard deviations and sums for amount paid by service categories using Proc Means for before and after six months of exam. The Macro "Word_it" uses ODS to output to Word and Macro "Excel_it" " uses ODS to output to Excel files. The formats that are predefined in the format statements are also transferred to the output and are called "Category" in the Demographic table. From the code several two demographic word tables and two excel tables are generated, some of the tables are shown below.

The an example of word tables created for Frequencies are:

Demographics for six months after:

Variable	Category	Frequency
Sex	1=Male	26
	2=Female	16
Race	1=White	22
	2=Af. American	15
	3=Other	5
Catcode	1	2
	4	2
	8	2
	10	25
	13	11

An example of Excel tables created for amount paid is :
Amount paid by Catcode for Services six months after

Catcode	Frequency	Percentage	Total Amount	Mean Amount	Standard Deviation
1	2	4.762	72.69	36.35	1.38
4	2	4.762	188.48	94.24	0
8	2	4.762	118	59	1.41
10	25	59.524	599	23.96	1.02
13	11	26.19	720	65.45	86.18
TOTAL	42	100	1698.17	40.43	48.01

The Code:

```
/*************************************************************************

PROGRAM NAME: SUGI28.sas

AUTHOR: Shabnam Mehra

DATE CREATED:   12/12/2002

PROJECT NAME:   SUGI PAPER

PROJECT DESC.:  To develop a code to automate the process
from data input to reports.

INPUT:    sugi_d1.mdb, Data2.sas

OUTPUT:  Word and Excel tables

PGM. FLOW DESC:
 1)First Generate SAS databases and clean it for the projects
 2)Put standard formats on the data
 3)Merge the database by a common ID and

INSTRUCTIONS:
1)Change the Libname as needed.
 2) Make Sure all the formats in the two databases are standard
for variables

*************************************************************************/;
libname sugi "A:\Sugi28";

%let libname=ba;
%let inpw="*****";
%let indb="M:\smehra\sugi_d1.mdb";

options fmtsearch= (&libname..dt_formats);

/*-------------------------------------------------------------*\
 * Importing the access database to sas                       *
 * Then clean and put formats as need by project *
\*-------------------------------------------------------------*/

PROC IMPORT OUT=WORK.data1
          DATATABLE="Data"
          DBMS=ACCESS2000 REPLACE;
          dbpwd=&inpw.;
          DATABASE=&indb.;
RUN;

/*-------------------------------------------------------------*\
 * Macro to Put Formats and labels in                         *
 * Data One that was imported from Access                     *
\*-------------------------------------------------------------*/
%Macro D_FORMATS(libname, dataname);

   options nofmterr;
   data &libname..&dataname.;
   set &dataname.;
      plocexam=put(locexam,1.);
      ID=put(CSSN,9.);
      SEX=put(csex,1.);
      RACE=put(crace,1.);
      Ctype=put(certtype,1.);

      Label      sex='Sex'
                 race='Race'
                 ctype='Certification Type'
                 plocexam='Location of Exam'
                 Exm_Cnty='County of Exam'
                 Exm_date='Date of Exam'
                 DupNum= 'Duplicate Record';
      Keep   ID plocexam Sex Race Ctype dupnum Exm_cnty
             Exm_date;

   run;

   options fmtsearch= (&libname..dt_formats);

   data &libname..&dataname;
   set  &libname..&dataname.;
      format  sex $sexf. race $racef. ctype $ctypef. dupnum
              dupnumf.  plocexam $locexamf. ID $9.;

   Proc Sort ;
       By ID;
   run;

%mend D_FORMATS;


/*-------------------------------------------------------------*\
 * Macro to pull those who have an exam date in data one and  *
 * receive services in data two. Sorting by SSN before merging  *
\*-------------------------------------------------------------*/

%Macro Merge_it (afile1,libname,afile2);

   data State;
   set &afile1.;
      rename race=race_dt2
             gender=sex_dt2;
   run;

   Proc Sort data=state;
     by ID;
   run;


 /*-------------------------------------------------------------*\
  * Creating data with six months before and after date of exam  *
  * Merging claims by ID, to check if they had services 6 months *
  *  before and after date of exam                               *
 \*-------------------------------------------------------------*/

Data Combo ;
   Merge  &libname..&afile2.(in=indt2) state(in=inSW);
   by ID;
      if indt2=1 then output;
run;

Proc Sql stimer feedback;
   create table ComboBef as
      select *, (EXM_date-183)as Beg_date format =mmddyy10.
      from combo  ;

   create table &libname..ComboBef as
      select *
      from ComboBef
      where servdate between Beg_date and Exm_date ;

   create table ComboAft as
```

```
      select *,(EXM_date+183)as Aft_date format =mmddyy10.
      from combo ;

   create table &libname..ComboAft as
      select *
      from ComboAft
      where servdate between Aft_date and EXM_date ;

 drop table &afile1. &afile2 combo.;

 quit;

 %mend Merge_it;

/*-------------------------------------------------------------------*\
 * Develop a code to generate tables for N, Percentage, *
 * Mean , Median and std deviation for Amount Paid       *
\*-------------------------------------------------------------------*/

Proc SQL;
Title1 " Frequency Of Demographics";
Title2 " Before Exam Date ";

   Create Table sugi.Demo_Freq_bef
      (
      Var_type Char(25)label= "Demographic Variable",
      var_class Char(25)label= "Class Variable",
      Count_N   Num  Label="Frequency"
      )
   ;

Title2 " After Exam Date ";
   Create Table sugi.Demo_Freq_aft
      (
      Var_type Char(25)label= "Demographic Variable",
      var_class Char(25)label= "Class Variable",
      Count_N   Num  Label="Frequency"
      )
   ;
quit;


%Macro Freq_it (libname,afile,var1,Freq_Tb);

    Proc freq data=&libname..&afile. noprint;
      tables &var1.  / nocum out=&var1.freq;
      run;

/*-------------------------------------------------------*\
 * Now joining the tables to get one output  *
\*-------------------------------------------------------*/
Proc Sql;
   Insert into &libname..&Freq_tb.
   Select "&var1.",
            &var1.,
            count
   from &var1.freq;
quit;

%mend Freq_it;

 /*-------------------------------------------------------------------*\
  * Develop a Macro to generate tables for N, Percentage, *
  * Mean , Sum and Standard Deviation for Amount Paid    *
 \*-------------------------------------------------------------------*/
%Macro Analyse_it (libname,afile,var1,var2,var1freq,amt_paid);
/*-------------------------------------------------------------------------*\
 * Using Proc means to generate a table containing Mean, Sum *
 * and Std. Deviation for Amount Paid by defined Categories      *
\*-------------------------------------------------------------------------*/

Proc  means  data= &libname..&afile. noprint;
   class &var1. ;
   Var &var2.;
   Output out=tempamt1  mean=Mean_amt sum= Sum_amt
            stddev=Std_amt ;
   title "Cost per Service utilization before Exam Date ";
run;

/*-------------------------------------------------------------*\
* Putting Formats and labels on the output table  *
\*-------------------------------------------------------------*/
Data tempamt1;
 set tempamt1;
   Mean_amt=Round(Mean_amt,.01);
   Sum_amt=Round(Sum_amt,.01);
   Std_amt= Round(Std_amt,.01);
   label Mean_amt= "Mean ofAmount Paid"
   Sum_amt="Sum of Amount Paid"
   Std_amt= "Standard Deviation for Amount Paid";
   Rename _Freq_ = Count;
   if _type_=0 and catcode="" then catcode="TOTAL";

run;


/*-------------------------------------------------------------------*\
* Now joining the tables of Frequency and Means to get one *
* Table using SQL                                          *
\*-------------------------------------------------------------------*/

Proc Sql;
Title1 " Frequency, Mean, Median and Standard deviation of ";
Title2 " Amount Paid by Catcode";

   Create table amt_paid as
      select a.catcode ,b.count , a.percent as pct_all
label='Percentage of All',
           b.catcode, b.sum_amt,b.mean_amt, b.std_amt

      from &var1freq. a full join tempamt1 b
      on a.catcode= b.catcode
      order by catcode;
      ;
   quit;

   data &libname..&amt_paid.;
   set  amt_paid;
      if catcode=" " then do;
            catcode="TOTAL" ;
            pct_all=100;
      end;
   Proc Sort ;
      by &var1.;
   run;
%mend analyse_it;

/*-------------------------------------------------------------*\
 * Develop a Macro to output the tables into Excel  *
 *  and Word using ODS statements                   *
\*-------------------------------------------------------------*/

%Macro Word_it(Table,libname,Sql_tb);
   ods rtf file="a:sugi28\&Table..rtf";
      proc print data=&libname..&SQL_tb. noobs;
      run;
   ods rtf  close;
%mend Word_it;


%Macro Excel_it(Table,libname,Sql_tb);
   ods csv file="a:sugi28\&Table..csv";
      proc print data=&libname..&SQL_tb. noobs;
      run;
   ods csv close;
%mend Excel_it;
```

```
/*---------------------- ---------------------------------*\
 * Calling in the Macros to run the process *
\*-------------------------------------------------------*/
%D_FORMATS (sugi, data1);
%Merge_it (Data2,sugi, data1);

%Freq_it(sugi,Combobef,sex, Demo_Freq_bef);
%Freq_it(sugi,Combobef,race, Demo_Freq_bef );
%Freq_it(sugi,Combobef,catcode,Demo_Freq_bef);
%F_Out_it (Demo_FQ_bef,Sugi,Demo_Freq_bef);
%Analyse_it(sugi,ComboBef, catcode,amtpaid,catcodefreq
,amt_bef);
%Excel_it (amt_bef,Sugi,amt_bef);

%Freq_it(sugi,ComboAft,sex, Demo_Freq_aft);
%Freq_it(sugi,ComboAft,race, Demo_Freq_aft );
%Freq_it(sugi,ComboAft,catcode,Demo_Freq_aft);
%Word_it (Demo_FQ_aft,Sugi,Demo_Freq_aft);
%Analyse_it(sugi,ComboAft, catcode,amtpaid,catcodefreq
,amt_aft);
%Excel_it (amt_aft,Sugi,amt_aft);
```

Until this code was developed at the data center I used general SAS code without SQL to complete the same task and exported the tables using Proc export® to produce the same results. Developing this code was a challenge, however once the code had been developed it has proved itself indispensible. This paper highlights only descriptive and basic statistics are automated. However any kind of statistics can be automated using Macro statements to build it and Proc Sql ®comes in as a handy tool to combine multiple statistics tables to generate one table of output which can then be outputted to Excel or word or even html using ODS Statements. Macro language in combination with proc SQL® and ODS is definitely an indispensible tool for a programmer to accomplish the challenge in short time.

CONCLUSION

Obtaining reports of descriptive statistics from mutliple datasets of different foramts and layouts can be easily accomplished using the Proc SQL®, Proc Freq®, Proc Means® and ODS. Macro statements can then help to automate the process if these reports are produced repeatively. It is a simple and swift method for manipulating secondary data to a desirable format for answering research questions.

TRADEMARKS

SAS, Proc SQL, Proc Export, Proc import, Proc Freq and Proc Means is a registered trademark of SAS Institute Inc., Cary, NC, USA and other countries.
® Indicates USA registration.

REFERENCES

SAS Version 8.2, Cary , NC : SAS Institute Inc.

SAS Institute Inc. (1990) , *"SAS Language Reference, Reference Version 6, Frist Edition ,"* Cary , NC : SAS Institute Inc.

SAS Institute Inc. (1990) , *"SAS Procedures Guide, Version 6, Third Edition ,"* Cary , NC : SAS Institute Inc.

SAS Institute Inc. (1999) , *"SAS Macro Language: Reference, Version 8."* Cary , NC : SAS Institute Inc.

SAS Institute Inc. (2000) , *"SAS SQL ProcedureUser's Guide Version 8."* Cary , NC : SAS Institute Inc.

Delwiche L.D.and Slaughter S.J.(2000) *"The Little SAS Book," Second Edition*, Cary , NC : SAS Institute Inc.

Beam D. (2001)." Intoduction to Proc SQL®". *Proceedings of Twenty SixthAnnual SAS Users Group International Conference*,150-26.

Mehra S., et al. (2002). "SPAN SPAN AWAY! Creating One Unique Record for Overlapping Admissions and Discharges from Multiple Inpatient Hospital Stays." *Proceedings of Twenty Seventh Annual SAS Users Group International Conference*, 214-27.

ACKNOWLEDGEMENT

Special thanks to Diane Haynes and Rebecca Larsen at our Data Center for their input to develop the code. I would also like to thank to the rest of the data center team for their encouragement and assistance.

CONTACT INFORMATION

Shabnam Mehra
University of South Florida
13301 Bruce B. Downs Blvd., MDC 2612
Tampa, Florida 33612
Phone: 813-974-9315
e-mail: smehra@fmhi.usf.edu