Paper 126-27

Using SAS Macros to Create Automated Excel Reports Containing Tables, Charts and Graphs

Tugluke Abdurazak
Abt Associates Inc.
1110 Vermont Avenue N.W. Suite 610
Washington D.C. 20005-3522
Tel: 202-263-1821, Fax: 202-263-1839
tugluke_abdurazak@abtassoc.com

ABSTRACT

Excel reports are always welcomed by the client because they are user-friendly, easy to access and easy to design. Therefore, many clients request reports to be created as Excel files with pre-designed formats. In most cases, the Excel report templates contain tables, charts, graphs or even pictures. Usually, an Excel report is created by using the output from SAS software, a SAS program outputs the result either as ASCII output or as an Excel file, programmers or other project staff cut, paste or type in the result to the pre-designed Excel report. This usually becomes a very time consuming and tedious process especially when the content of the reports is complex and when reports need to be created repeatedly. This program attempted to automate the process so that SAS analytical results are automatically output to the pre-defined Excel templates. Because the report had to be created repeatedly, the automation process saved a lot of time and money. The program is easy to modify to create reports for a variety of projects. This approach was greatly appreciated by SAS programmers, project managers and clients, and was successfully implemented for different reports.

Introduction

In last year, we have received a request from our client to track an ongoing survey by reporting statistics on key variables on a weekly base for a certain period of time. The client designed the report as an Excel file and would like us to produce the report as fast as possible based on a very tight budget. The whole process of producing reports based on pre-defined templates could be considered as three stages. In the first stage, project manager, staff and SAS programmers developed a strategy to solve the problem. After serious discussion, we decided to automate the process so that SAS analytical results were automatically inserted into the pre-defined Excel reports. There are two major reasons for making this decision. 1. The report format is created by out client, we expect frequent changes to be made during the process. 2. The reports have to be created every week until the survey is completed. To manually create such a huge number of reports is expensive and is not feasible. In the second stage, SAS programmers and project staffs work closely to develop plans to create reports. A SAS programmer writes SAS Macros to automatically output initial analytical results to Excel spreadsheets. Excel macros automatically link the results to pre-defined Excel reports. In the final stage, the project manager and SAS programmers quality check the results to make sure that the reports that are produced accurately fulfil the requirements from the client.

Process Illustration

Excel Template Preparation

An Excel template has to be created before running the program. The template contains multiple pages (spreadsheets or Excel tables). The main page is the exact copy of the Excel report specified by the client. It may contain tabulations, charts, graphs, descriptions and/or maps. Other pages are to be used for storing output from the SAS program; for clarification purposes, I call them "data tabs." Once the SAS program is executed, the SAS procedure outputs the statistical result as SAS data set. Different SAS macros manipulate these data sets and create new tables according to the specification. These tables are then exported directly to the data tabs in an Excel template. The programmer then carefully links each data tab to the relevant tables in the main page. The very first report is the most time-consuming one. It is important that this be correct, because any error at this stage will affect future processing. After the first report is created, checked and approved by the client, subsequent production of reports becomes extremely easy.

The SAS Program

The complete program consists of the following three sections:

Section I: Assigns paths to hold the final report, defines libraries containing SAS data sets and creates temporary data sets to be used for the process. All the temporary SAS data sets in the working directory will be deleted. The title lines also are defined in this section. This is the only section that requires user interaction and is limited to the changes on the folder and file specifications only; it therefore is very easy to implement.

Section II: SAS Macros to be used repeatedly during the process. There are three macros written for this program:

1. %WORDS: Counts the number of words in a character string and returns that value. This macro is mainly used for counting the number of variables in a variable list for further processing. For example, the following statement will assign the number seven to the macro variable N:

```
%LET VARLIST = VAR1 VAR2 VAR3 VAR4 MALE
FEMALE GENDER;
%LET N = %WORD(&VARLIST );
```

2. %TAB1LOOP: Automatically produces frequencies and percentages of a list of binary variables, merges and concatenates the results to form a table according to pre-defined formats and outputs it as a SAS data set which is ready to be output to Excel. This macro also takes care of different issues related to data processing. For example, when all data values are missing for a certain variable, the macro creates an empty row or column. When the data set is empty, the SAS macro still outputs an empty table shell.

3. %EXCELOUT: Outputs the SAS data set to the specified position in the worksheet in a pre-defined Excel file. As an example, suppose that you have a SAS data set named 'POP' with 3 rows and 7 columns (3 by 7 matrix). Execution of the macro code %EXCELOUT(SDS=pop, XLSSHT=data,XLSF=c:\report.xls, ROW=2,COL=5) will automatically output the SAS data set 'pop' to the worksheet 'data' always starting from row 2 and column 5 in the Excel file named 'report.xls' under 'c:\' directory. Please note that the number of rows and the number of columns in the SAS data set are automatically counted by the macro.

Section III: Produces output data sets and outputs them to the Excel report by executing macros. Automatically generates the final report, renames it according to the date of execution and sends the report to the specified directory.

The complete program is shown below with a detailed description of every section and each step by using macro comments '%*'. The program has proven to be an excellent device for tracking an on-going survey. It automatically generates Excel reports for the client and project managers on a weekly basis for a long period of time. The program could be run against actual databases in the survey center and could output results directly to the client-specified location. This process eliminates the tedious and repetitive work of pulling data, running analyses, creating and editing Excel reports, and proves to be a very efficient way to produce Excel reports.

```
/*******************************************************************************************/
/* PROGRAM     : SUGI27_REPORT.SAS                                                         */
/* SAS Version : 8.0 or 6.12                                                               */
/* Tested Under: Windows NT                                                                */
/* PURPOSE     : This program uses SAS Macros to Create Automated Excel Reports            */
/*               Containing Tables, Charts and Graphs. The program is written for SUGI27   */
/*               conference.                                                               */
/* AUTHOR      : Tugluke Abdurazak                                                         */
/* COMPANY     : ABT ASSOCIATES                                                            */
/* ADDRESS     : 1110 Vermont Avenue,N.W., Suite 600, Washington D.C. USA 20005-3522       */
/* PHONE/FAX   : 202 263-1821 / 202 263-1839                                               */
/* E-MAIL      : tugluke_abdurazak@abtassoc.com                                            */
/*******************************************************************************************/

*SECTION I: Assign paths, set up options and create temporary datasets ;
*--- Set up options ;
OPTIONS /* SYMBOLGEN */ NOXWAIT MPRINT;
*---Folder contain SAS programs/INPUT statement/Files necessary for SAS process ;
%LET PROGPATH = D:\ABT_PROJECT\SUGI\SASPROG ;
*---Folder for holding SAS data sets ;
%LET SDSPATH = D:\ABT_PROJECT\SUGI\SASDATA ;
*---Folder for outputting Reports or other document ;
%LET FPATH = D:\ABT_PROJECT\SUGI\DOCS ;
*--- Final report folder ;
%LET RPTPATH = D:\ABT_PROJECT\SUGI\REPORT ;
*--- Excel file name. This should the name of Excel report pre-defined by the client;
%LET EXLFILE = SUGI_REPORT_TEMPLATE.XLS ;
*--- Set up SAS library ;
LIBNAME LIB "&SDSPATH" ;

*--- Delete temporary data sets in the Work library ;
PROC DATASETS LIBRARY=WORK KILL;
QUIT ;

*--- Titles appear on each page in the output window ;
TITLE1 "Pneumococcal Pneumonia and Influenza Immunization Remeasurement Survey &SYSDATE" ;

*--- Create temporary data for processing ;
*--- Create binary variables for the columns in the report ;
DATA SUGI (KEEP= DUM STATE SAMPST AGEGP RACE TZONE WEIGHT SEX AGEGP S3Q2
                 MALE FEMALE FLU PNEU WWW MOREPH RES_IN RES_OUT INCOME
                 EASTERN CENTRAL MOUNTAIN PACIFIC ALASKA);
  SET LIB.ALL_FNL ;
  *NOTE: Data coding statements excluded from the paper.;
RUN;

*--- Temporary data set to be used for merging other columns for Age Group table ;
DATA AGEGP_TEMP;
  DO AGEGP = 0 TO 3;
    OUTPUT ;
  END ;
RUN;

*--- Temporary data set to be used for merging other columns for Race Group table ;
DATA RACE_TEMP;
  DO RACE = 0 TO 6;
    OUTPUT ;
  END ;
RUN;

*SECTION II: Create macros ;

*---Macro %WORDS counts the number of words in a string ;
%MACRO WORDS(STRING);
  %* Parameter STRING contain the character string ;
  %LOCAL COUNT WORD;
  %* Macro variable COUNT contains a running total of the number of words ;
  %* in parameter STRING, and WORD temporarily contains each word identified ;
  %* in parameter STRING ;
  %LET COUNT=1 ;
```

```
  %LET WORD=%QSCAN(&STRING,&COUNT,%STR( ));
  %* The %LET statement selects the first word in STRING using the %QSCAN ;
  %* function. The %QSCAN function quotes the value, including mnemonic operators ;
  %* such as AND and OR, so that the value does not cause problems for the implicit ;
  %* %EVAL function in the %DO %WHILE condition. The %QSCAN function divides words ;
  %* only by blanks;
  %DO %WHILE(&WORD NE) ;
  %* %DO %WHILE loop executes as long as a word is present ;
    %LET COUNT=%EVAL(&COUNT+1);
    %* The %LET statement increments the value of COUNT by 1;
    %LET WORD=%QSCAN(&STRING,&COUNT,%STR( ));
    %* The %LET statement selects the next word ;
  %END;
  %* Since the value of COUNT is incremented before the %LET statement selects ;
  %* a new word, the value of COUNT is always one more than the number of words ;
  %* in the string. Therefore, the %VALUE function returns the number of words ;
  %* in the string. ;
  %EVAL(&COUNT-1)
%MEND WORDS;

*---TAB1LOOP macro is used for creating TABLE for estimated population;

%MACRO TAB1LOOP(SDS,WT=WEIGHT,VAR=) ;
  %* Declare local macro variables ;
  %LOCAL I CURVAR ;
  %* %WORDS count the number of variables in &VARLIST ;
  %DO I =1 %TO %WORDS(&VARLIST) ;
  %* Assign the ith variable name to the macro variable CURVAR ;
  %LET CURVAR = %SCAN(&VARLIST, &I) ;
  %* Data set TEMP contains a subset of &SDS where &CURVAR = 1 only ;
  DATA TEMP;
    SET &SDS;
    WHERE &CURVAR=1 ;
  RUN;

  %* Check to see if the data set TEMP is empty ;
    DATA _NULL_ ;
      IF 0 THEN SET TEMP NOBS=COUNT;
      CALL SYMPUT('NUM',LEFT(PUT(COUNT,8.)));
      STOP;
    RUN;
    %* If the data set TEMP is NOT empty, output frequency ;
    %IF &NUM NE 0 %THEN
      %DO;
        PROC FREQ DATA=&SDS ;
          TITLE3 "Frequency of &VAR by &CURVAR" ;
          TABLES &VAR*&CURVAR / OUT=&VAR._&CURVAR ( WHERE=(&CURVAR=1) RENAME=(COUNT=CNT_&CURVAR
                                                      PERCENT=PCT_&CURVAR));
          WEIGHT &WT ;
        RUN;
        %* Check to see if the output data set is empty;
          DATA _NULL_ ;
            IF 0 THEN SET &VAR._&CURVAR NOBS=COUNT;
            CALL SYMPUT('NUM2',LEFT(PUT(COUNT,8.)));
            STOP;
          RUN;
        %* If the output is empty, create a data set containing variables with missing values ;
        %IF &NUM2 EQ 0 %THEN
          %DO;
            DATA &VAR._&CURVAR;
              SET &VAR._TEMP;
               CNT_&CURVAR  = . ;
                PCT_&CURVAR  = . ;
                     RUN;
         %END;
         %ELSE
           %* If the output is not empty, merge it with temporary data set ;
           %DO;
              DATA &VAR._&CURVAR;
                SET &VAR._&CURVAR;
                 DROP &CURVAR ;
```

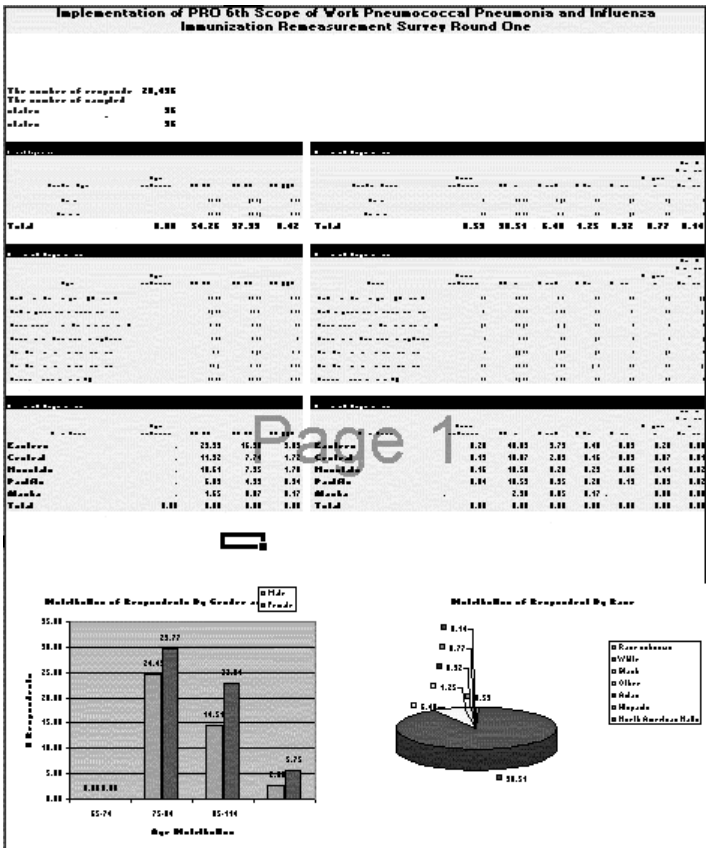
```
            RUN;
            DATA &VAR._&CURVAR;
              MERGE &VAR._TEMP (IN=A)
                    &VAR._&CURVAR;
              BY &VAR ;
              IF A;
            RUN;
          %END;
        %END;
      %ELSE
      %* else if the data set TEMP is empty, output a data set containing variables ;
       %* with missing values and put a NOTE on the LOG window. ;
        %DO;
          %PUT %STR();
          %PUT NOTE: ***** THERE ARE NO RECORDS WITH &CURVAR = 1 ***** ;
          %PUT NOTE: ***** OUTPUT AN EMPTY DATA SET &VAR._&CURVAR ***** ;
           %PUT %STR();
          DATA &VAR._&CURVAR ;
            SET &VAR._TEMP;
            CNT_&CURVAR  = . ;
            PCT_&CURVAR  = . ;
          RUN;
        %END;
       %* Create data set &VAR by merging all columns by macro variable &VAR ;
       %IF &I=1 %THEN
        %DO;
           DATA &VAR ;
             SET &VAR._&CURVAR ;
           RUN;
        %END;
       %ELSE
        %DO;
           DATA &VAR ;
            MERGE &VAR (IN=A)
             &VAR._&CURVAR  ;
            BY &VAR ;
           RUN;
        %END;
  %END ;

  %* Output transpose of data set &VAR. ;
  PROC TRANSPOSE DATA=&VAR OUT=&VAR._TR (DROP=_LABEL_) ;
    ID &VAR ;
  RUN;
  %* Split the data set to two data sets containing Counts and Percentages respectively ;
  %* this data set will be output to Excel report by macro %EXCELOUT ;
  DATA &VAR._CNT
       &VAR._PCT ;
    SET &VAR._TR ;
        IF SUBSTR(TRIM(LEFT(_NAME_)),1,3)= 'CNT' THEN OUTPUT &VAR._CNT;
          ELSE IF SUBSTR(TRIM(LEFT(_NAME_)),1,3)= 'PCT' THEN OUTPUT &VAR._PCT ;
  RUN;
%MEND;

*---Macro EXCELOUT Output SAS data set to Excel spreadsheet in the report template;

%MACRO EXCELOUT(SDS=,XLSSHT=,XLSF=, ROW=,COL= ) ;

  %* %MACRO statement initiates the Macro EXCELOUT *;
  %* Parameter SDS is the SAS data set name to be exported to Excel. Parameter XLSSHT specifies *;
  %* the name of the spreadsheet in the Excel file. Parameter XLSF indicates the location and  *;
  %* name of the Excel file. *;
  PROC CONTENTS DATA=&SDS NOPRINT OUT=CNT ;
  RUN;
  PROC SORT DATA=CNT ;
    BY VARNUM ;
  RUN;
  %* PROC CONTENTS is used to output a SAS data set CNT which will be used to count the number *;
  %* of rows and columns in the SDS. One of the reasons for sorting CNT VARNUM become obvious  *;
  %* in the following step *;
  PROC SQL NOPRINT;
```

```
    SELECT NAME
    INTO: VARS SEPARATED BY ' '
    FROM CNT ;
    SELECT COUNT(DISTINCT NAME)
    INTO: COLS SEPARATED BY ' '
    FROM CNT ;
    SELECT NOBS
    INTO: ROWS
    FROM CNT
    WHERE VARNUM = 1;
  QUIT;
  %* PROC SQL is used to create macro variables VARS (the number of variables),ROWS the number *;
  %* of rows) and COLS (the number of columns in the SDS. Please note the NOBS in the CNT is *;
  %* equivalent to the number of rows in SDS. Sorting CNT by VARNUM played a role here. *;
  %* Output to Excel report &EXLFILE ;
  OPTIONS NOXWAIT NOXSYNC ;
  X "&XLSF" ;
  %* X command is used to issue a host system command to open a pre-defined Excel file. ;
  %* Wait 5 seconds for Excel to execute ;
  DATA _NULL_ ;
    X=SLEEP(5);
  RUN ;
  %* DATA _NULL_ is used to give time for Excel to execute. SLEEP(10) function would allow *;
  %* Excel file 10 seconds to open up. ;
  FILENAME TEMP DDE "EXCEL|&XLSSHT.!R&ROW.C&COL.:R%TRIM(%EVAL(&ROWS+&ROW-1))C%TRIM(%EVAL(&COLS+&COL))" ;
  DATA _NULL_ ;
    SET &SDS ;
    FILE TEMP ;
    PUT &VARS ;
  RUN ;
  %* DDE option is used with FILENAME statement to read data from SAS and write to Excel ;
  %* For more information about DDE (Dynamic Data Exchange.), see "DDE Syntax within the SAS System" ;
  %* in the "Using Dynamic Data Exchange" chapter in SAS Companion for the Microsoft Windows Environment. ;
  FILENAME CMDS DDE 'EXCEL|SYSTEM' ;
  DATA _NULL_ ;
    FILE CMDS ;
    PUT '[SAVE()]' ;
    PUT '[QUIT()]' ;
  RUN ;
  %* CMDS and DDE option is used with FILENAME statement to issue command SAVE() for saving *;
  %* Excel file and QUIT() to close Excel file. ;
  %* %MEND statement ends a macro %EXCELOUT *;
%MEND EXCELOUT ;

*SECTION III: Generate output data sets and export Excel files ;

*--- Create a table containing counts of States and # Observations, ;
*--- the table will be output to the header portion of the Excel report ;
PROC SQL;
  CREATE TABLE HEADER
  AS
  SELECT COUNT(DISTINCT STATE) AS N_ST,
         COUNT(DISTINCT SAMPST)AS N_SAMPST,
         COUNT(*) AS N_OBS
  FROM SUGI;
QUIT;

*--- Create AGE and RACE tables ;
%LET VARLIST = MALE FEMALE FLU PNEU WWW MOREPH RES_IN RES_OUT INCOME
               EASTERN CENTRAL MOUNTAIN PACIFIC ALASKA ;

*--- Execute Macro TAB1LOOP for Age Group report ;
%TAB1LOOP(SUGI,WT=DUM,VAR=AGEGP) ;
*--- Execute Macro TAB1LOOP for Race Group report ;
%TAB1LOOP(SUGI,WT=DUM,VAR=RACE) ;
*---Output SAS data sets to Excel report &EXLFILE ;
%EXCELOUT(SDS=HEADER,XLSSHT=DATA,XLSF=&FPATH\&EXLFILE,ROW=3,COL=1 ) ;
%EXCELOUT(SDS=AGEGP_CNT,XLSSHT=DATA,XLSF=&FPATH\&EXLFILE,ROW=5,COL=1 ) ;
%EXCELOUT(SDS=RACE_CNT,XLSSHT=DATA,XLSF=&FPATH\&EXLFILE,ROW=5,COL=6 ) ;
%EXCELOUT(SDS=AGEGP_PCT,XLSSHT=DATA,XLSF=&FPATH\&EXLFILE,ROW=21,COL=1 ) ;
%EXCELOUT(SDS=RACE_PCT,XLSSHT=DATA,XLSF=&FPATH\&EXLFILE,ROW=21,COL=6 ) ;
```

```
*--- Copy the report to Report folder ;
X "COPY &FPATH\&EXLFILE &RPTPATH" ;
X "CD &RPTPATH" ;

*---Wait for 5 seconds for DOS prompt;
DATA _NULL_ ;
  X=SLEEP(5);
RUN ;

*--- Rename the report SUGI_REPORT_DATE.XLS where DATE is the date when ;
*--- this program is executed.;

X "RENAME &EXLFILE SUGI_REPORT_&SYSDATE..XLS" ;
*---Wait for 5 seconds for DOS prompt;
DATA _NULL_ ;
  X=SLEEP(5);
RUN ;

*--- Exit DOS ;
X "EXIT" ;

/*******************************END OF PROGRAM*********************************************/
```

The Final Report

Below is one of the final Excel reports created by the program. It contains tables of analytic results, charts and graphs of population estimates as well as descriptions. All these are automatically updated in real time according to the actual response from the survey center.



Summary

The SAS program developed in this process uses SAS Macros, SAS procedures and Excel software to create Excel reports with pre-defined formats. Since the process is automated, it requires very little user input and therefore could be run by anyone with a basic knowledge of SAS and Excel software. The program has been tested numerous times and has been used for creating about 200 reports for clients. It should be pointed out, however, that the program has only been tested under Windows NT and Windows 2000. Microsoft Excel has to be installed correctly into the user's system. Therefore, the program is not guaranteed to work with other operating systems.

Acknowledgements

My sincere appreciation to Jeffrey Phillips for encouragement and support in finishing up this study, and to Linda Piccinino for editing drafts of this paper and taking her precious time.

References

1. Professional SAS Programming Secrets by Rick Aster & Rhena Seidman. ISBN 0-07-913095-X (pbk.)
2. SAS Guide to Macro Processing, Version 6, Second Edition (SAS Institute Inc.). ISBN 1-55544-382-6
3. SAS Procedure Guide, Version 8 (SAS Institute Inc.). ISBN 1-58025-482-9
4. SAS Macro Facility Tip & Techniques, Version 6, First Edition (SAS Institute Inc.). ISBN 1-55544-605-1