Paper 011-30

Automating the Drudgery Away: Using Macros and ODS to Produce (Almost) Complete Reports

Phil Rhodes, Baylor University, Waco, TX

ABSTRACT

The mission of the Office of Institutional Research and Testing (IRT) at Baylor University is to conduct research in order to provide information which supports institutional planning, policy, and decision making. As part of that mission, IRT produces approximately 75 research-related reports each year, along with several hundred ad hoc reports about prospective students, enrolled students, faculty, and staff. Six of the most important of these reports are the annual profiles of the student body, produced each fall. Previously, these reports were generated on a mainframe and the output printed. The data for the tables in each report was then manually entered into a Microsoft Excel template. This was a tedious and error-prone process, requiring five staff members many hours of data entry and proofreading. This paper will describe the steps taken to streamline the production of these reports using the SAS Macro Language and ODS. The current process takes one person less than an hour to produce the reports and requires minimal proofreading due to automatic generation of the report tables.

SAS products used include Base SAS, the SAS Macro Language, and ODS. Basic knowledge of PROC TABULATE, the SAS Macro Language, and ODS is assumed.

INTRODUCTION

The mission of the Office of Institutional Research and Testing (IRT) at Baylor University is to conduct research in order to provide information which supports institutional planning, policy, and decision making. As part of that mission, IRT maintains three different data warehouses (running on SAS® software) with student, human resources, and financial data. Additionally, IRT produces approximately 75 official reports each year, along with several hundred ad hoc reports about prospective students, enrolled students, faculty, and staff. Six of the most important of these reports are the annual profiles of the student body, produced each fall.

The annual profiles produced by IRT contain biographic, demographic, test score, and academic data about new freshmen, undergraduate students, new transfers, graduate students, law students, and seminary students. Each profile begins with a short narrative section discussing the current year and previous year, followed by a selection of tables displaying the characteristics of the relevant group of students. In the past, the data for these tables was generated using mainframe SAS, printed, and then retyped into production-quality Microsoft Excel templates. This tedious and error-prone process took three to four hours per profile (including careful proofreading).

During the spring and summer of 2003, Baylor University converted to a new student system. This involved major changes to the underlying student data, as well as rewrites to virtually every SAS program in the IRT library. Coinciding with this conversion was an evaluation of many of the office processes. It was decided that the process of producing profiles could and should be streamlined to save time.

PROFILE REPORT STRUCTURE

The first step was to examine the existing programs used to generate the profile data and to 'profile' the profiles for common segments. It quickly became apparent that there was a large overlap in the profiles that could be exploited using SAS Macros. The profiles had the following items in common:

- comparison of two years of data – this fall and last fall
- same set of demographic tables
- two basic table types - one for biographic/demographic information and one for test score information
- some information required for narrative not available directly from the tables

The primary differences in the profiles were in test score and academic information. The profiles differed in the number and type of test scores used in the report. Certain profiles contained academic information missing in other profiles - the freshmen and transfer profiles contained a table of academic programs, for example, while the undergraduate and graduate profiles did not (that information is contained in a separate, more comprehensive, report). The freshmen, transfer, and undergraduate profiles also had a table on high school class quartile, which was meaningless for the graduate, seminary, and law profiles.

After studying the profiles, it was decided that trying to create one large macro program with all of the logic to create each profile would be too difficult to maintain. Thus, the profiles would still be generated by separate programs. However, common code blocks would be rewritten as parameterized macros and placed in a separate program to simplify updating and maintenance. Additionally, this separate program would contain all of the formats unique to the profiles. This program could then be used via %INCLUDE in any profile (or other program).

After more analysis, the common code blocks were rewritten into the following macros:

- %PROCTAB - a parameterized PROC TABULATE for non-numeric data
- %PROCTAB_TEST - a parameterized PROC TABULATE for numeric (primarily test score) data
- %PROFILE_SETUP - a macro to extract data from frozen files and create new variables as needed
- %PROFILE_DEMOGRAPH - a macro to produce the common demographic section of a profile

THE TABLE MACROS

The two table macros were the first written, as they were easiest to code and test - not to mention that the rest of the programs were dependent on them! The code for macro %PROCTAB follows:

```
%macro proctab(DSNAME,VAR,SORTORD,VARFMT,LABEL,MISSFLAG);
   %* Macro to generate the proc tabulates used by most of the tables;
   %* Parameters:
    * DSNAME - source dataset
    * VAR - variable to report on
    * SORTORD - sort order, either ascending or descending, or blank to use data order
    * VARFMT - format of variable
    * LABEL - text to place in upper left corner of table
    * MISSFLAG - set to missing to include missing values, blank otherwise
    *
    * Note that CELWIDTH and LBLWIDTH are global macro variables referenced here
    ;

   proc tabulate data=&DSNAME format=6.
            style=[background=white cellwidth=&CELWIDTH] &MISSFLAG;
      class &VAR /  &SORTORD style=[background=white cellwidth=&LBLWIDTH];
      class term / descending style=[background=white];
      classlev &VAR term / style=[background=white];
      format &VAR &VARFMT term $term.;
      keylabel pctn='%' n='N';
      keyword all n pctn / style=[background=white];
      table &VAR=" " all,term=" "*(n*f=comma6. pctn<&VAR all>*f=6.1) /
            box = [label="&LABEL" style=[background=white cellwidth=&LBLWIDTH]]
            misstext='--';
   run;
%mend proctab;

ods proclabel="Demographics - Gender";
%proctab(profile_data, gender, descending, $gender., Gender,);
```

Note that this macro is not truly a 'generic' PROC TABULATE - coding all of the TABULATE options would have complicated the macro considerably. Instead, it is fine tuned for a particular set of reports. The last two lines of code show how the macro would typically be called, and Figure 1 shows the resulting table:

Gender	Fall 2004		Fall 2003	
	N	%	N	%
Male	1,104	39.6	1,001	37.4
Female	1,681	60.4	1,677	62.6
All	2,785	100.0	2,678	100.0

Figure 1: %PROCTAB Output

The %PROCTAB_TEST macro is very similar to %PROCTAB. It has an additional parameter, the numeric analysis variable ANLVAR, as well as a VAR statement in the PROC TABULATE and changes to the TABLE statement. Differences from the previous macro are highlighted in grey:

```
%macro proctab_test(DSNAME,CLASSVAR,SORTORD,VARFMT,ANLVAR,LABEL,MISSFLAG);
   %* Macro to generate the proc tabulates used for the quartiles and means of test
         scores;
```

```
    %* Parameters:
     * DSNAME - source dataset
     * VAR - variable to report on
     * SORTORD - sort order, either ascending or descending, or blank to use data order
     * VARFMT - format of variable
     * ANLVAR - analysis variable
     * LABEL - text to place in upper left corner of table
     * MISSFLAG - set to missing to include missing values, blank otherwise
     *
     * Note that CELWIDTH and LBLWIDTH are global macro variables referenced here
     ;
    proc tabulate data=&DSNAME format=6.
             style=[background=white cellwidth=&CELWIDTH] &MISSFLAG;
       class &CLASSVAR / &SORTORD style=[background=white cellwidth=&LBLWIDTH];
       class term / descending;
       classlev &CLASSVAR term / style=[background=white];
       var &ANLVAR;
       format &CLASSVAR &VARFMT term $term.;
       keylabel pctn='%' n='N';
       keyword all n pctn mean / style=[background=white];
       table &CLASSVAR=" ",term=" "*&ANLVAR=" "*mean=" "*f=4. /
             box = [label="&LABEL"
             style=[background=white cellwidth=&LBLWIDTH]]
             misstext='--';
    run;
%mend proctab_test;
```

The call to %PROCTAB _TEST

```
%proctab_test(quartiles,vartype, ,$20.,value,Interquartile Range,missing);
```

produces the output in Figure 3:

Interquartile Range	Fall 2004	Fall 2003
25th Percentile	1090	1085
75th Percentile	1280	1270

Figure 2: %PROCTAB_TEST Results

THE SETUP MACRO

All of the profiles required data to be extracted from frozen files. The macro %PROFILE_SETUP extracts the data, creates additional variables, and sets up macro variables needed by the rest of the profile program. It takes three parameters: a profile type, the current fall term, and the previous fall term:

```
%macro profile_setup(TERM1, TERM2, PROFTYPE);
   %* Parameters:
    *    TERM1 - first term of interest
    *    TERM2 - second term of interest
    *    PROFTYPE - type of profile to produce
    *       FR - new freshmen
    *       TR - new transfers
    *       UG - undergraduates
    *       GR - graduate students
    *       SM - seminary
    *       LW - law
   ;
```

The bulk of the macro consists of two data steps: a DATA _NULL_ step that creates multiple macro variables, some of which depend on the profile type, and a data step to extract the data for the profile. Here is a portion of the DATA _NULL_ step that creates a macro variable, VARLIST1, containing a list of variables common to all profiles:

```
/* Variables to keep for all profiles */
call symput('VARLIST1','term gender age ethnicity ethord ethnicn ethnord foreign
                        fornord full_part_time_ind religion_code relord home_state
                        citizen_natn citizen_code unit_all unitord curr_att_hrs_sem
                        class_adjusted classord class');
```

Depending on the profile type, other macro variables are created:

```
select(proftype);
    when('FR')
    do;  /* Set up for freshman profile */
       call symput('TITLE1',"Profile of First-Time Freshmen, "
                        || trim(term1word) || " and " || trim(term2word));
       /* Where clause to select records - NEEDS TRAILING SEMICOLON */
       call symput('WHERECLS','where new_student_type = "F";');
       /* Additional variables to keep for profile */
       call symput('VARLIST2','high_school_quarter program_p1 progord
                        ed_goals_primary ed_goals_ord satv satm satt sathigh
                        acte actm actr acts actc acthigh new_student_type');
       /* Output file name */
       call symput('OUTFILE',trim(filepath) || "profile-freshman-" ||
                                              trim(term2) || ".pdf");
    end;  /* When FR */
```

Once a profile type is selected, macro variables are set up to hold:

- the report title
- a WHERE clause used to subset the data
- any additional variables required for the profile type
- an output file name

This approach made it much simpler to add new profiles to the list if necessary (in fact, two additional profile subjects were added after the macros were initially written, taking much less time than writing them from scratch).

Once the macro variables are created, the data can be extracted from the frozen files:

```
/* Get data to generate profile */
   data profile_data;
      set census.st_census_&TERM1 (keep=&VARLIST1 &VARLIST2)
          census.st_census_&TERM2 (keep=&VARLIST1 &VARLIST2);
      %* Note:  where clause macro variable should include a final semicolon;
      &WHERECLS
```

Note the use of the macro variables from the parameters (TERM1 and TERM2), as well as those created in the previous DATA _NULL_ step.

THE DEMOGRAPHIC REPORT MACRO

The macro to produce the common demographic section of the report, %PROFILE_DEMOGRAPH, was written next. It takes one parameter, indicating the profile type, and prints all of the demographic tables. A portion of the code is shown below:

```
%macro profile_demograph(proftype);
      /* Title for page, not individual proc */
      title1 "&TITLE1";
      title2 "Demographic Information";

      ods proclabel="Demographics - Gender";
      %proctab(profile_data, gender, descending, $gender., Gender,);

      ods proclabel="Demographics - Age";
      proc sort data=profile_data;
         by ageord age;
      run;

      %* Determine format for age variable;
      %if "&PROFTYPE" = "FR" %then
```

```
            %let AGEFMT = age1fmt.;
         %else %if "&PROFTYPE" = "UG" or "&PROFTYPE" = "TR" %then
            %let AGEFMT = age2fmt.;
         %else
            %let AGEFMT = age3fmt.;

         %proctab(profile_data, age, %str(order=data), &AGEFMT, Age, missing);

              ...

   %mend profile_demograph;
```

In %PROFILE_DEMOGRAPH , liberal use was made of

```
   ods PDF startpage=now;
```

to group short tables together on the same page.

Once all of the macros were written and tested, individual profile programs were written. All six had the same basic structure:

- initial macro variable definitions, including terms, the profile type, and the file location
- a %INCLUDE to bring in the program with the profile macros
- a call to %PROFILE_SETUP to extract the data
- a call to %PROFILE_DEMOGRAPH to print the demographic tables to PDF
- calls to %PROCTAB and %PROCTAB_TEST to print the rest of the tables to PDF
- additional SAS code to print a report with data for the narrative (sent to OUTPUT)

The programs produce the major part of each report, the tabular data. As mentioned above, there is also a narrative section that includes additional information not readily available in the tables (the number of states and countries reprresented, for example). The narrative section is still manually produced in Microsoft Word. For now…

To view the completed reports, go to http://www.baylor.edu/irt/ and click on one of the profiles listed in the callout box on the right side of the page.

CONCLUSION

IRT recognized several benefits from the conversion of the profiles to use macros. First and foremost, the profiles accurately reported the data without meticulous proofreading and verification. Second, the new programs yielded an immense time savings. The reports were generated by one staff member in less than an hour instead of taking five staff members 12-18 hours. Proofreading also took much less time, about 15 minutes per report, instead of the hour or more needed to meticulously check the old reports. Finally, updating the profiles to add a new table or section is much easier with most of the code in one place.

CONTACT INFORMATION

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

Phil Rhodes
Baylor University
One Bear Place #97032
Waco, TX 76798
Email: Phillip_Rhodes@baylor.edu