A Unique Approach to Create Custom Reports by Leveraging the Strengths of SAS® and Excel®

Amy O. Wilkerson, Barbara S. Bibb, Brett P. Anderson, Mai T. Nguyen • RTI International, Research Triangle Park, NC

Survey projects often require custom reports to allow project staff to monitor production as well as produce various statistics from the collected data. At RTI, we’ve come up with a unique approach for creating custom reports for our projects by leveraging the strengths of SAS® and Excel®. In SAS® we use PROC SQL to select and when necessary aggregate data. After processing the data in SAS®, results are sent to Excel® for reporting and graphics.

1. Introduction

A client requested an Excel® report showing the most up-to-date information from an online survey that would be generated automatically overnight. The survey itself was programmed in Hatteras®, a proprietary software program developed by RTI International. The report used both data collected by the survey as well as data coded by the systems. This process takes advantage of the features of Excel®, SQL Server®, and SAS®. A worksheet is built using the formulas required for the desired report; and then data is extracted using PROC SQL. This data is output to separate Excel® worksheets; the first worksheet, containing the formulas, picks up the data from the data storage worksheets for the final calculations and formatting.

An example of an Excel® report is shown in Table 1 below. Two graph output examples are shown in Figures 1 and 2.

3. Steps to Create Excel Reports

Step 1. Set Up Lib Names and Blank Excel® Worksheets to Hold and Summarize Codes

Libname in '\...\Patient Dataset';
Libname out '\...\Reports';
Libname REMS ODBC complete='DRIVER=SQL Server;
SERVER=rtpwsql12;UID=xxx;PWD=xxx;DATABASE=xxx;';
Libname xls excel '\...\Patient_Survey_Response_Report.xlsx';
Libname codes excel '\...\patient_codes.xls';
%let sumstat_codes = 'sumstat_codes$';
%let gender_codes = 'gender_codes$';
%let age_codes = 'age_codes$';
%let time_codes = 'time_codes$';
%let mode_codes = 'mode_codes$';

Step 2. Add a Section to Re-Initialize the Data

* Remove existing data first;
proc datasets lib=xls nolist;
delete sumstat_report gender_report age_report time_report mode_report;
run;

You can see the report tab displayed with formulas and the data storage tabs associated with the report in Table 2 below.

Table 2. Formulas in Excel

The remainder of the SAS® program repeats the process just described for the other variables of interest, inserting and storing the data on the support worksheets. The final step is updating the formulas on the Patient Survey Response Report worksheet to refer to the counts of variables on the other worksheets.

In addition to outputting data as an Excel spreadsheet, the output results can also be displayed as a graph:

4. Conclusion

5. Discussion

6. Alternative Graphic Output