Paper 208-213

A Unique Approach To Create Custom Reports By Leveraging the of SAS® and Excel®

Amy O. Wilkerson, Barbara Bibb, Mai Nguyen and Brett Anderson, RTI International

ABSTRACT

Survey projects often require custom reports to allow project staff to monitor production as well as 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.

In our paper, we will present a few sample reports, program codes and the detailed explanations of how these reports were created.

INTRODUCTION

The 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®, 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.

The report output in Excel® is shown in Table 1 on the next page.

Patient Survey Response Report

Overall	Count	% of Complete
Number Completed Patient Surveys	**201**	**100%**
Self interview (web)	125	62%
Interviewer (telephone)	76	38%
Completed by Gender	**Count**	**% of Complete**
Male	28	14%
Female	173	86%
Total	**201**	**100%**
Completed by Age	**Count**	**% of Complete**
40 years old or younger	5	2%
41 – 50 years old	12	6%
51 – 60 years old	60	30%
61 – 70 years old	67	33%
71 years old and older	57	28%
Total	**201**	**100%**
Completed by Time Taking Medication	**Count**	**% of Complete**
Less than 2 months	25	12%
2 months to less than 4 months	53	26%
4 months to less than 6 months	40	20%
6 months to 1 year	43	21%
More than 1 year	40	20%
Total	**201**	**100%**

Table 1

DETAILS

Below is an example of the code needed to create multiple Excel® worksheets that capture data from a previously generated SAS® dataset. For this particular report, five separate Excel®

worksheets were generated which contained the numbers of different summary status codes (sumstat_codes), counts of patients' gender (gender_codes), categorical age ranges (age_codes), the time using the medication in question (time_codes), and the mode in which the survey was completed (mode_codes – telephone or self-administered).

```
libname in '\\...\Patient Dataset';
libname out '\\...\Reports';
libname REMS ODBC complete='DRIVER=SQL
Server;SERVER=rtpwsql12\pubprod;UID=xxxx;PWD=xxxx;DATABASE=xxxx;';

libname xls excel '\\...\Patient_Survey_Response_Report.xlsx';
libname codes excel '\\...\patient_codes.xls';

%let sumstat_codes = 'sumstat_codes$'n;
%let gender_codes = 'gender_codes$'n;
%let age_codes = 'age_codes$'n;
%let time_codes = 'time_codes$'n;
%let mode_codes = 'mode_codes$'n;
```

Once the sheets were created, commands to wipe out the old data were added, since this report was run on a nightly basis. The new daily data is deleted, but the report worksheet with its formulas remains intact.

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

The next step was to write PROC SQL procedures to categorize each data point in which the client was interested. The first PROC SQL generates a count of different summary status codes (incomplete, complete, or ineligible). A left join is necessary here in order to associate the sum stat codes with the SQL® data point. The values for sum stat are 3-digit numeric codes (for example, 102, 250 or 295).

```
* Get count of sumstat codes;
proc sql;
      create table temp as
      select c.*,
             case
                  when r.caseid is not null then 1
                    else 0
               end as sumstat_count
      from codes.&sumstat_codes c left join in.rems r
        on c.code = input(r.sumstat, 8.);
quit;
```

The next PROC SQL statement adds up and displays the summary status codes in a particular worksheet of the report:

```
proc sql;
      create table xls.sumstat_report as
      select code, sum(sumstat_count)
      from temp
      group by code;
quit;
```

The previous code generates a count of the completed surveys and mode of administration and adds a percent of the total surveys at the far right:

Overall	Count	% of Complete
Number Completed Patient Surveys	**201**	**100%**
Self interview (web)	125	62%
Interviewer (telephone)	76	38%

Table 2

As you can see in Table 2, there were a total of 201 patients who completed the interview, either over the phone with an interviewer or by self-administration directly on the website. 76 patients, or 38% of this group, completed the interview over the telephone, and 125 patients, or 62%, completed the interview on the web.

The Excel® spreadsheet containing the working formulas is shown in Table 3:

Overall	Count	% of Complete
Number Completed Patient Surveys	**=B8+B9**	**=C8+C9**
Self interview (web)	=mode_report!B3	=(mode_report!B2)/B7
Interviewer (telephone)	=mode_report!B2	=(mode_report!B3)/B7
Completed by Gender	**Count**	**% of Complete**
Male	=gender_report!B2	=(gender_report!B2)/B7
Female	=gender_report!B3	=(gender_report!B3)/B7
Not answered	=gender_report!B4	=(gender_report!B4)/B7
Total	**=B11+B12+B13**	**=C11+C12+C13**
Completed by Age	**Count**	**% of Complete**
40 years old or younger	=age_report!B2	=(age_report!B2)/B7
41 – 50 years old	=age_report!B3	=(age_report!B3)/B7
51 – 60 years old	=age_report!B4	=(age_report!B4)/B7
61 – 70 years old	=age_report!B5	=(age_report!B5)/B7
71 years old and older	=age_report!B6	=(age_report!B6)/B7
Not answered	=age_report!B7	=(age_report!B7)/B7
Total	**=B16+B17+B18+B19+B20+B21**	**=C16+C17+C18+C19+C20+C21**
Completed by Time Taking Medication	**Count**	**% of Complete**
Less than 2 months	=time_report!B2	=(time_report!B2)/B7
2 months to less than 4 months	=time_report!B3	=(time_report!B3)/B7
4 months to less than 6 months	=time_report!B4	=(time_report!B4)/B7
6 months to 1 year	=time_report!B5	=(time_report!B5)/B7
More than 1 year	=time_report!B6	=(time_report!B6)/B7
Not answered	=time_report!B7	=(time_report!B7/B7)
Total	**=B24+B25+B26+B27+B28+B29**	**=C24+C25+C26+C27+C28+C29**

Report | age_report | time_report | sumstat_report | mode_report | gender_report

Table 3

You can see the report tab displayed with formulas and the data storage tabs associated with the report.

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.

Amy Overby Wilkerson
RTI International
3040 Cornwallis Road
Research Triangle Park, NC 27709
919-541-8021