PharmaSUG 2015 - Paper PO12

Utilizing SAS® for Cross-Report Verification in a Clinical Trials Setting

Daniel Szydlo, Fred Hutchinson Cancer Research Center, Seattle, WA
Iraj Mohebalian, Fred Hutchinson Cancer Research Center, Seattle, WA
Marla Husnik, Fred Hutchinson Cancer Research Center, Seattle, WA

ABSTRACT

Large scale efficacy clinical trials require frequent and diligent monitoring of study data that includes the generation of Data Safety Monitoring Board (DSMB) reports. The importance of data accuracy, frequency of generation, and sheer length of DSMB reports combine to make their verification crucial and time-consuming. In an effort to improve the accuracy and efficiency of DSMB report production we developed procedures utilizing SAS® to automate a number of verification checks. This paper presents the framework and the SAS® tools we developed for verifying DSMB reports for a phase III efficacy trial that are generalizable to other research reports. The tools we developed to support DSMB report production include the following: SAS® programs that create standardized results datasets for the generation of tables, listings, and figures (TLFs); driver files that determine the verification checks to be performed; SAS® programs that perform verification checks on results datasets; and HTML-formatted output that clearly show whether and where errors occurred. We present specific examples of our tools using mock data from a phase III clinical trial that utilize SAS® facilities such as the Macro language and Output Delivery System (ODS).

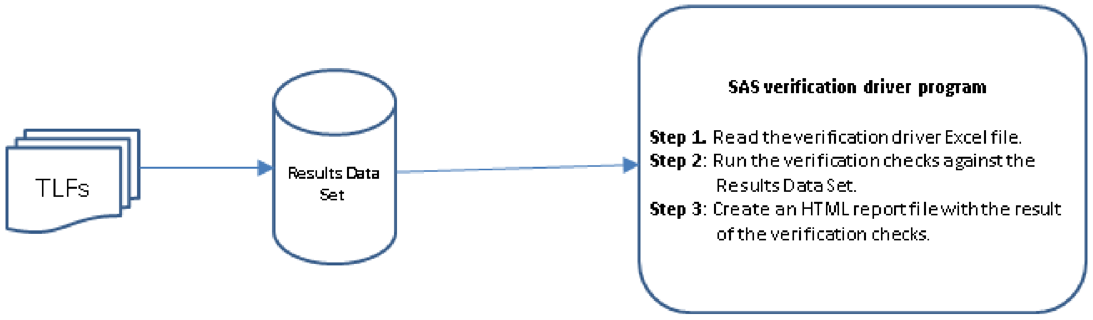
INTRODUCTION

Study Monitoring Committees (SMC) provide oversight to study conduct and Data Safety Monitoring Boards (DSMB) provide monitoring of safety and efficacy data for large scale clinical trials. Statistical and Data Management Center (SDMC) staff prepare reports for SMCs and DSMBs that involve the generation of tables, listings, and figures (TLFs) on approximately a biannual basis throughout study follow-up. Given the importance of safety and efficacy monitoring in large scale clinical trials, the accuracy of the data in these reports is extremely important and, therefore, requires thorough review and verification prior to completion of a report. However, manual verification checks can be time-consuming and prone to errors, especially when checking large numbers of TLFs. Thus, developing a system in SAS® to automate the verification process is undoubtedly worthwhile and aids in increasing the accuracy and efficiency of this burdensome task.

Our SDMC team was involved in a large-scale clinical trial that required production of ten SMC and DSMB reports from August, 2012 through November, 2014. The latest SMC report included 61 tables, 4 listings, and 26 figures totaling 111 pages. The latest DSMB report consisted of 128 tables, 7 listings, and 43 figures totaling 627 pages. Since the SMC and DSMB meetings generally occurred within a month of one another, our team was responsible for the production of approximately 750 total pages of TLFs during any given reporting period. Many of the TLFs contained identical information such as the following: title statements, footnotes indicating the date of TLF generation, and summaries of total enrollment. In addition, during report production, corrections to formatting, changes in the layout of the tables, and additions or subtractions to the TLFs were made. As a result, our team was faced with performing manual verification of a large quantity of data for each SMC or DSMB report. Examples of the checks we conducted are as follows: unexpected data errors, confirmation that enrollment numbers were identical across all tables, consistency across table titles, consistency across footnotes, and SAS® logs for potential issues. Given this, we developed an automated cross-report verification system using SAS®, and were able to complete the verification tasks with a great deal of efficiency, accuracy and generalizability for future projects. Note that the examples shown throughout this paper use mock data and reference dates only.

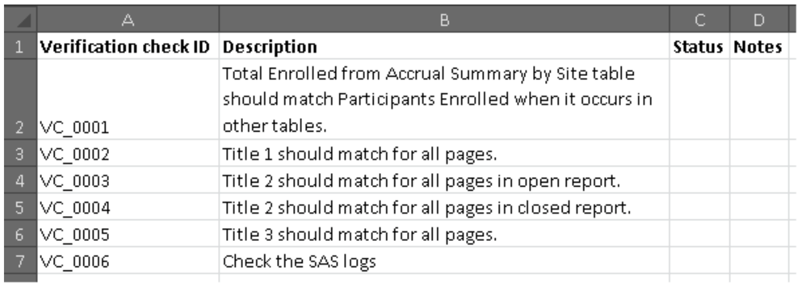
GENERAL FRAMEWORK

The tables and listings are created by constructing derived data sets and output using PROC REPORT. Our aim is to have the tables appear uniform throughout the reports and so we construct derived data sets using a standardized structure. A standardized structure of the data sets allows us to concatenate them into a master data set that consists of all of the data that was used to create the SMC or DSMB report. This process is shown in Display 1.


TLFs
Results Data Set
SAS verification driver program
Step 1. Read the verification driver Excel file.
Step 2: Run the verification checks against the Results Data Set.
Step 3: Create an HTML report file with the result of the verification checks.

Display 1. Framework Flow Diagram

Once the master data set, referred to as the Results Data Set, is created we perform our verification checks. To do this we first create a spreadsheet, referred to as the Excel Verification Driver file that contains four columns: “Verification check ID”, “Description”, “Status”, and “Notes”. “Verification check ID” contains a unique identifier for each verification check that we want to perform, “Description” briefly summarizes the aims of the check, and “Status” and “Notes” are initially left blank and later filled by the verification check SAS® programs. It is also possible to record known exceptions in the “Notes” column. An excerpt of a sample Excel Verification Driver file is shown in Display 2.



	A	B	C	D
1	Verification check ID	Description	Status	Notes
2	VC_0001	Total Enrolled from Accrual Summary by Site table should match Participants Enrolled when it occurs in other tables.		
3	VC_0002	Title 1 should match for all pages.		
4	VC_0003	Title 2 should match for all pages in open report.		
5	VC_0004	Title 2 should match for all pages in closed report.		
6	VC_0005	Title 3 should match for all pages.		
7	VC_0006	Check the SAS logs		

Display 2. Excel Verification Driver File

For each of the unique verification check IDs we wrote a SAS® program that uses the Results Data Set to perform the check. Each of these individual programs generate two macro variables, &VC_<IDnumber>_status and &VC_<IDnumber>_notes. After all of the individual SAS® verification check programs execute and the macro variables are stored, we then run a macro we created called %read_htm that writes the information from the Excel Verification Driver to an HTML file and adds the values from &VC_<IDnumber>_status and &VC_<IDnumber>_notes to the row of the relevant verification check ID number. The end result is an HTML page that informs us of whether or not the verification check passed or failed along with descriptive information regarding the cause of failure if it exists.

CREATION OF RESULTS DATA SET

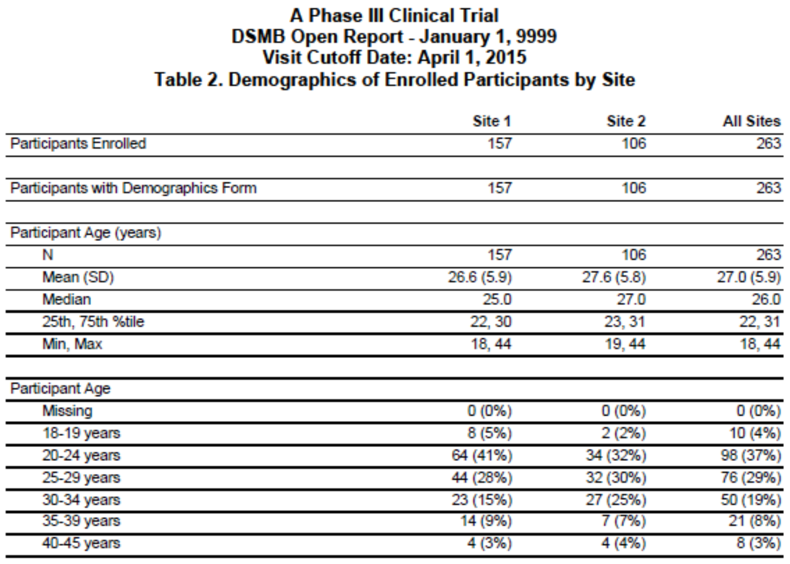
For the purposes of performing verification checks, the Results Data Set must follow a standardized format and contain enough information to make every element of the TLFs uniquely identifiable. We primarily use internal macros to create the data sets that are used in conjunction with PROC REPORT to create the tables and listings contained in the SMC and DSMB reports, but any method of data set creation that satisfies the above criteria could work as well.

Variables in the Results Data Set include the following:

- prgmid (Program ID): Directory location and file name of SAS® program that is used to generate the data set containing the data element.
- dataset_created_from: Unique identifier for name of standardized data set (some prgmid correspond to multiple dataset_created_from values).
- number_of_titles: Number of titles used for TLF.
- title1-title10, footnote1-footnote10: Values of title1-title10 and footnote1-footnote10 for a TLF, if used.
- rowvar (Row variable description): Row heading in the table.
- colvar(Column variable description): Column heading in the table.

- repvar (Report variable): Value of a cell in a table (or data element in a figure)

In order to see how these variables correspond to an actual table, Display 3 shows a sample demographics table using mock data.



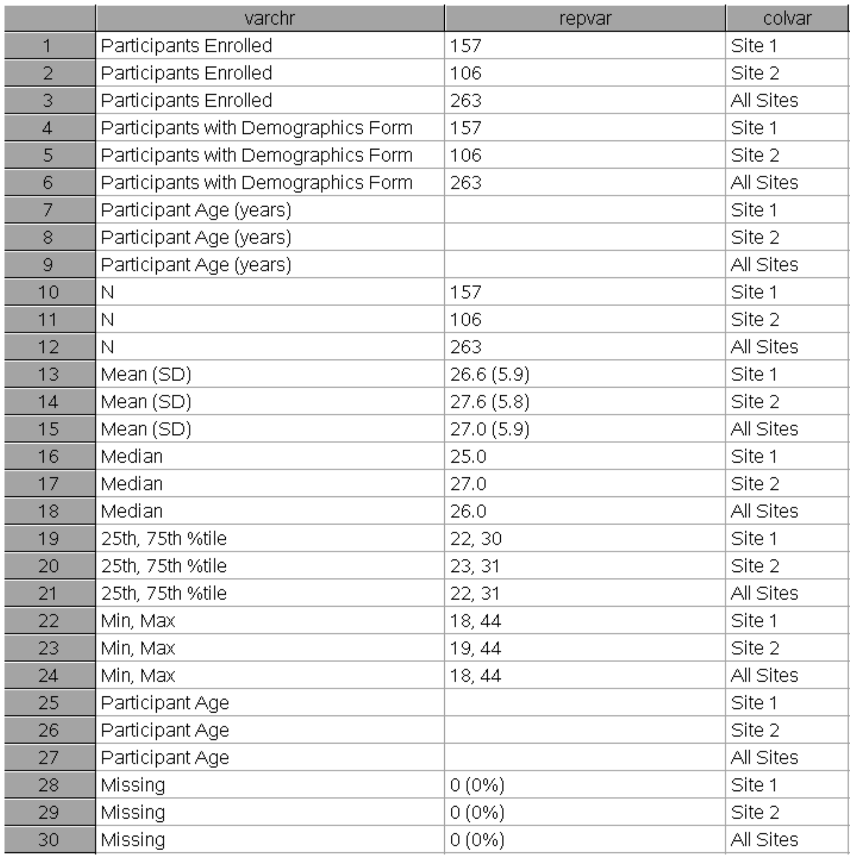
A Phase III Clinical Trial
DSMB Open Report - January 1, 9999
Visit Cutoff Date: April 1, 2015
Table 2. Demographics of Enrolled Participants by Site

	Site 1	Site 2	All Sites
Participants Enrolled	157	106	263
Participants with Demographics Form	157	106	263
Participant Age (years)			
N	157	106	263
Mean (SD)	26.6 (5.9)	27.6 (5.8)	27.0 (5.9)
Median	25.0	27.0	26.0
25th, 75th %tile	22, 30	23, 31	22, 31
Min, Max	18, 44	19, 44	18, 44
Participant Age			
Missing	0 (0%)	0 (0%)	0 (0%)
18-19 years	8 (5%)	2 (2%)	10 (4%)
20-24 years	64 (41%)	34 (32%)	98 (37%)
25-29 years	44 (28%)	32 (30%)	76 (29%)
30-34 years	23 (15%)	27 (25%)	50 (19%)
35-39 years	14 (9%)	7 (7%)	21 (8%)
40-45 years	4 (3%)	4 (4%)	8 (3%)

Display 3. Sample Demographics Table from DSMB Report

Referring to the sample demographics table, prgmid takes on the name of the SAS® program used to generate this table and the directory path where the program is located. Here, that value is /open/ t_dem_site_blind.sas. The value for dataset_created_from is also not displayed in the table itself but is important for identifying which of the many TLFs are being verified. In this case the dataset_created_from variable equals demout_site, and will correspond 1-1 with the prgmid value. If, however, t_dem_site_blind.sas had created a table with multiple pages dataset_created_from would differentiate by being equal to demout_site1, demout_site2, and so forth for each page of the table. The values of number_of_titles and number_of_footnotes are 4 and 0, respectively, with title1 = A Phase III Clinical Trial, title2 = DSMB Open Report – January 1, 9999, title3 = Visit Cutoff Date: April 1, 2015, and title4 = Table 2. Demographics of Enrolled Participants by Site.

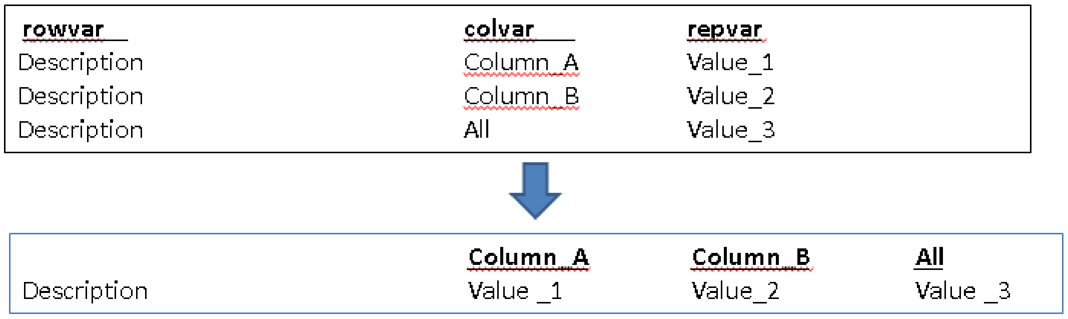
Comparing the first row of the table shown in Display 3 to the data set shown in Display 4, we see that the first row of the data set, with rowvar = Participants Enrolled, colvar = Site 1, and repvar = 157, corresponds to the Participants Enrolled row and Site 1 column in the table. Similarly, the second row of the data set, with rowvar = Participants Enrolled, colvar = Site 2, and repvar = 106, corresponds to the Participants Enrolled row and Site 2 column in the table.



	varchr	repvar	colvar
1	Participants Enrolled	157	Site 1
2	Participants Enrolled	106	Site 2
3	Participants Enrolled	263	All Sites
4	Participants with Demographics Form	157	Site 1
5	Participants with Demographics Form	106	Site 2
6	Participants with Demographics Form	263	All Sites
7	Participant Age (years)		Site 1
8	Participant Age (years)		Site 2
9	Participant Age (years)		All Sites
10	N	157	Site 1
11	N	106	Site 2
12	N	263	All Sites
13	Mean (SD)	26.6 (5.9)	Site 1
14	Mean (SD)	27.6 (5.8)	Site 2
15	Mean (SD)	27.0 (5.9)	All Sites
16	Median	25.0	Site 1
17	Median	27.0	Site 2
18	Median	26.0	All Sites
19	25th, 75th %tile	22, 30	Site 1
20	25th, 75th %tile	23, 31	Site 2
21	25th, 75th %tile	22, 31	All Sites
22	Min, Max	18, 44	Site 1
23	Min, Max	19, 44	Site 2
24	Min, Max	18, 44	All Sites
25	Participant Age		Site 1
26	Participant Age		Site 2
27	Participant Age		All Sites
28	Missing	0 (0%)	Site 1
29	Missing	0 (0%)	Site 2
30	Missing	0 (0%)	All Sites

Display 4. Rowvar, colvar, repvar from the Demographics Table

Other rowvar, colvar, and repvar value combinations contain the information necessary to create the entire demographics table. A general representation of this process is illustrated in Display 5, with the first box corresponding to the form that the data takes in the Results Data Set, and second box corresponding to the output as it appears in the table that is used for the SMC or DSMB report.



rowvar	colvar	repvar
Description	Column_A	Value_1
Description	Column_B	Value_2
Description	All	Value_3

	Column_A	Column_B	All
Description	Value _1	Value_2	Value _3

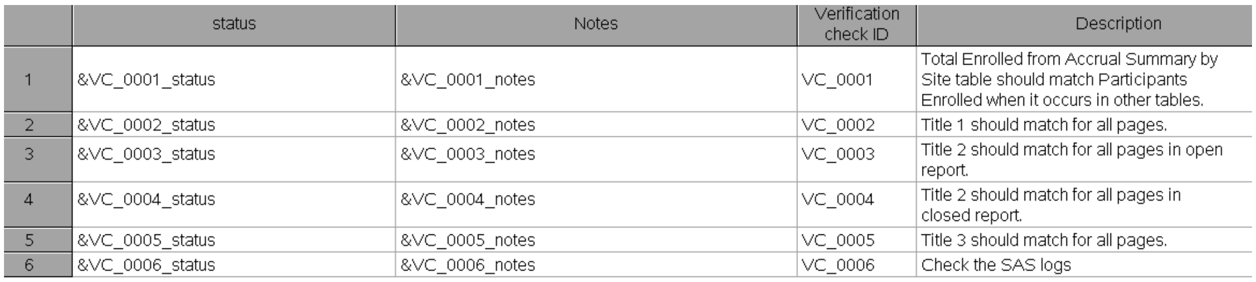
Display 5. Structure of Results Data Set

DRIVERS FOR VERIFICATION CHECKS AND HTML OUTPUT

The Excel Verification Driver file, called verification_checks.xlsx in the example below, defines the list of verification checks to be run for each report. The SAS® driver program, read_verification_checks.sas, creates an HTML template file. An example of this code is shown below.

```
PROC IMPORT out=checks datafile="Verification_checks.xlsx" dbms=excelcs replace;
    server="sassvc";
run;

DATA temp;
    attrib Status length=$20 format=$20. Notes length=$256 format=$256;
    set checks(where=(^missing(Verification_check_id)));
    Status = strip(Status)||'&'||strip(Verification_check_id)||'_status';
    Notes  = strip(Notes)||'&'||strip(Verification_check_id)||'_notes';
run;

ODS HTML file= "Verification_template.htm";
PROC REPORT data=temp headline headskip nowd ls=256
           center style(report)=table[frame=below rules=rows];
    column obs Verification_check_ID Description Status Notes;
    define Obs   / 'Obs' center;
    define Verification_check_ID / 'Verification Check ID' center order
           style(column)=[cellwidth=2in just=center];
    define Description / 'Description' flow width=50 style(column)=[cellwidth=5in
           just=left] ;
    define Status /  'Status' flow style(column)=[cellwidth=1in just=center];
    define Notes / 'Notes' style(column)=[cellwidth=4in just=left]  flow;
    compute obs;
     cntl+1;
     obs = cntl;
    endcomp;
run;

ODS HTML close;
```

The DATA step that creates the data set entitled “temp” reads the Verification_check_id and Description variables from the Excel Verification Driver file and adds macro variables to the Status and Notes variables. Display 6 shows the work.temp data set that results from the above code.



	status	Notes	Verification check ID	Description
1	&VC_0001_status	&VC_0001_notes	VC_0001	Total Enrolled from Accrual Summary by Site table should match Participants Enrolled when it occurs in other tables.
2	&VC_0002_status	&VC_0002_notes	VC_0002	Title 1 should match for all pages.
3	&VC_0003_status	&VC_0003_notes	VC_0003	Title 2 should match for all pages in open report.
4	&VC_0004_status	&VC_0004_notes	VC_0004	Title 2 should match for all pages in closed report.
5	&VC_0005_status	&VC_0005_notes	VC_0005	Title 3 should match for all pages.
6	&VC_0006_status	&VC_0006_notes	VC_0006	Check the SAS logs

Display 6. Data Set work.temp

This data set is then written to an HTML template file using the PROC REPORT statement. The macro values in the status and notes variables remain unresolved at this time.

VERIFICATION CHECKS

Once the HTML template has been created we carry out the verification checks themselves. We use separate SAS® programs for each of the checks detailed in the Excel Verification Driver file. Each program has a specific verification aim, such as confirming that the enrollment totals match across all TLFs. The program accomplishes this by comparing all occurrences of the relevant data element within the Results Data Set and determining whether the values of those cells satisfy the conditions of the check. During this process the values for the macro variables &VC_<IDnumber>_status and &VC_<IDnumber>_notes are created and stored.

Below we discuss three examples in detail: enrollment numbers verification check, titles consistency verification check, and the SAS log file verification check.

ENROLLMENT NUMBERS VERIFICATION CHECK

Most tables included in the SMC and DSMB reports contain the total number of participants enrolled in the study. If discrepancies exist between enrollment numbers on different tables it could be an indication that a table was generated using outdated data, that there is an error in the code that generates a particular table, or it could be the result of other causes. The sheer number of times that total enrollment is presented makes this check ideal for automation.

Total enrollment generally appears in tables in a row with the heading of “Participants Enrolled”, and occurs in a column corresponding to All Arms, All Sites, or All Countries, depending on whether the table is presented by arm, site, or country. Using the Results Data Set, called work.save_all in the example below, we identify the relevant observations for this check by sub-setting to observations where the rowvar value equals “Participants Enrolled” and the colvar value equals whichever values we use to represent All Arms/Sites/Countries. In the demographics table from Display 3, site has been formatted such that 1=’Site 1’, 2=’Site 2’, and 999=’All Sites’.

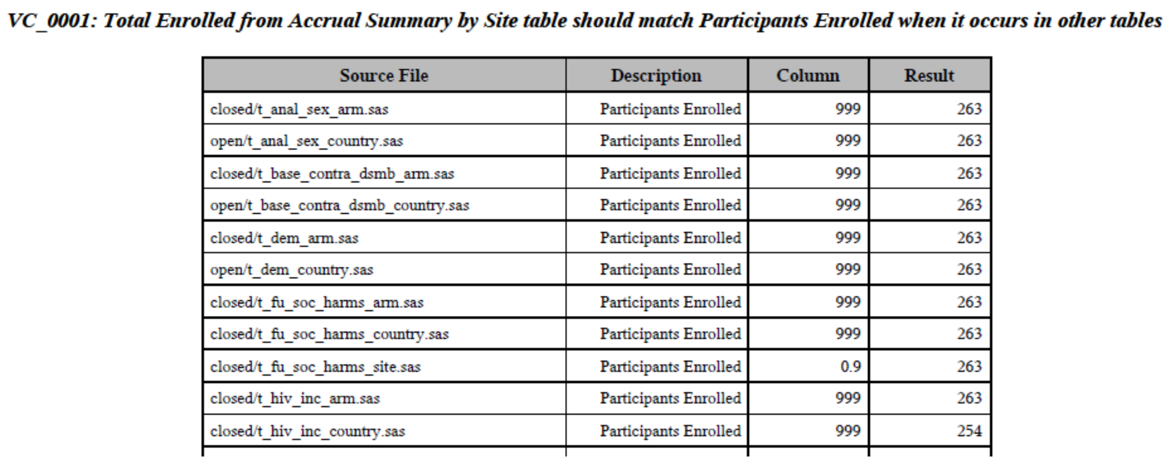
```
DATA VC_0001;
    set work.save_all;
    if upcase(strip(VARCHR)) =: 'PARTICIPANTS ENROLLED' and colvar = 999;
run ;
```

We then create a new variable called num_repvar as a numeric version of repvar, which is a character variable in the Results Data Set. This variable is used to perform the verification check, but first we need to create a pdf file summarizing our verification check, as shown in the code below.

```
ODS pdf file = "VC_0001.pdf" bookmarklist=hide notoc;
title "VC_0001: Total Enrolled from Accrual Summary by Site table should match
Participants Enrolled when it occurs in other tables";
PROC REPORT data=all nowd ls=256;
    column prgmid varchr colvar num_repvar;
    define prgmid /'Source File' display style(column)=[cellwidth=6in just=left];
    define varchr/'Description' display style(column)=[cellwidth=1.5in just=right];
    define colvar /'Column' display style(column)=[cellwidth=1in  just=right];
    define num_repvar /'Result' display style(column)=[cellwidth=1in  just=right];
run;
ODS pdf close;
```

For the check itself we use PROC SQL, as shown below, to compare the minimum and maximum values of the num_repvar variables; if the minimum and maximum are equal then all values must be identical. Likewise, if the minimum and maximum are equal then an %IF statement gives the &VC_0001_status macro variable the value of “Passed”, otherwise it gives the value of “Failed”. We also assign an HTML link containing a path to the pdf summary file, so that SDMC staff can easily access the file containing the summary. In this case the summary contains the source file name, description, column, and result of the “Participants Enrolled” values across all of the report tables. A value is also assigned to the &VC_0001_notes macro variable. For this check we desire a way to quickly see some of the enrollment numbers, so the note will display a concatenation of all the num_repvar values.

```
PROC SQL noprint;
    select min(num_repvar)- max(num_repvar) into :diff  from VC_0001;
    quit;
run;
%MACRO check;
    %global VC_0001_status VC_0001_notes;
    %let VC_0001_notes = &notes;
    %if &diff = 0 %then %let VC_0001_status=<A HREF="VC_0001.pdf">Passed</A>;
    %else %let VC_0001_status=<A HREF="VC_0001.pdf">Failed</A>;
%MEND check;
%check;
```



VC_0001: Total Enrolled from Accrual Summary by Site table should match Participants Enrolled when it occurs in other tables

Source File	Description	Column	Result
closed/t_anal_sex_arm.sas	Participants Enrolled	999	263
open/t_anal_sex_country.sas	Participants Enrolled	999	263
closed/t_base_contra_dsmb_arm.sas	Participants Enrolled	999	263
open/t_base_contra_dsmb_country.sas	Participants Enrolled	999	263
closed/t_dem_arm.sas	Participants Enrolled	999	263
open/t_dem_country.sas	Participants Enrolled	999	263
closed/t_fu_soc_harms_arm.sas	Participants Enrolled	999	263
closed/t_fu_soc_harms_country.sas	Participants Enrolled	999	263
closed/t_fu_soc_harms_site.sas	Participants Enrolled	0.9	263
closed/t_hiv_inc_arm.sas	Participants Enrolled	999	263
closed/t_hiv_inc_country.sas	Participants Enrolled	999	254

Display 7. Excerpt from summary file for VC_0001

In the excerpt of VC_0001.pdf shown in Display 7 we see that not all of the total enrollment numbers match. Most are 263, but the table generated by t_hiv_inc_country.sas has an enrollment total of 254. Due to this discrepancy the verification check fails and will cause &VC_0001_status to equal <A HREF="VC_0001.pdf">Failed</A>.

TITLES CONSISTENCY VERIFICATION CHECK

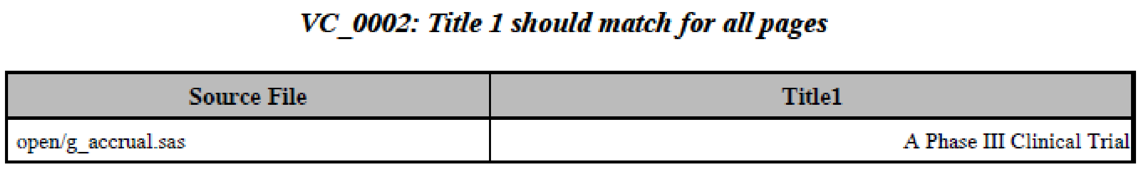
TITLE statements are another data element that are included across all TLFs in a report and should be similar or identical, making it appropriate for a verification check. Titles are included on every page of our DSMB and SMC reports and they provide information about the name of the study, the type of report, the date of the meeting, the date of data cutoff, and a TLF-specific description. The standard information is contained in the first three title lines. Referring to Display 3, the first line of the title is “A Phase III Clinical Trial”, and this title will be the same for all TLFs in the SMC or DSMB report. The second line reads “DSMB Open Report – January 1, 9999”, indicating that this table was generated for the Open portion of a DSMB meeting that is to be held on January 1, 9999, which is a placeholder date, in this case. This line can change depending on whether the TLF was generated for the DSMB Open, DSMB Closed, SMC Open, or SMC Closed reports. The third line, “Visit Cutoff Date: April 1, 2015” will be consistent across the Open and Closed reports.

We use four verification check SAS® programs to evaluate the title statements; VC_0002.sas checks all title1 statements for consistency, VC_0003.sas checks title2 statements from the Open version of the DSMB or SMC report, VC_0004.sas checks title2 statements for the Closed version, and VC_0004.sas checks all title3 statements. For VC_0002 we simply use PROC SORT along with nodupkeys to verify that all title1 values are equal.

```
PROC SORT data=work.save_all out=VC_0002 nodupkeys;
    by title1;
run;
```

The VC_0002.pdf summary displays the records from this VC_0002 dataset, with one row for every unique title statement in the report. This only includes one source file for each unique title statement, so if there is a discrepancy additional debugging is required to determine whether multiple programs are triggering the failure.

```
ODS PDF file = "VC_0002.pdf" bookmarklist=hide notoc;
TITLE "VC_0002: Title 1 should match for all pages";
PROC REPORT data=VC_0002 nowd ls=256;
    column prgmid title1;
    define prgmid /'Source File' display style(column)=[cellwidth=3in  just=left];
    define title1 /'Title1' display style(column)=[cellwidth=4in just=right];
run;
ODS pdf close;
```



VC_0002: Title 1 should match for all pages

Source File	Title1
open/g_accrual.sas	A Phase III Clinical Trial

Display 8. Summary PDF for VC_0002

Our verification check then simply counts the number of observations in the VC_0002 data set, with 1 observation leading to a pass and any other number of observations causing a failure. We have written a short macro program to accomplish this task, called %num_obs, as shown below.

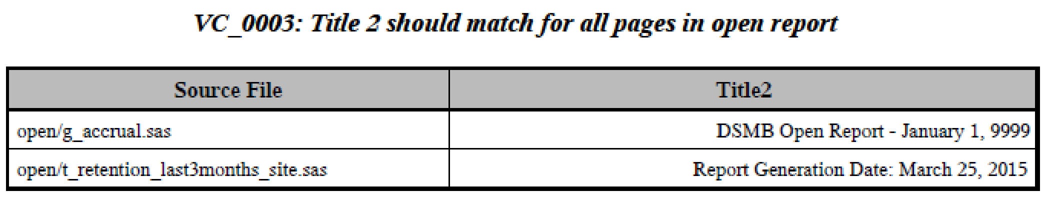
```
%MACRO num_obs (dsname);
%GLOBAL numobs;
DATA _null_;
    %if (%sysfunc(exist(&dsname))) %then %do;
           if 0 then set &dsname nobs=nobs1;
           call symput ('numobs',trim(left(put(nobs1,8.))));
    %end;
    %else %do;
           call symput ('numobs',"0");
    %end;
    stop;
run;
%MEND num_obs;
%num_obs(VC_0002);

%MACRO check;
    %global VC_0002_status VC_0002_notes ;
    %let VC_0002_notes = &notes;
    %if &numobs = 1 %then %let VC_0002_status=<A HREF="VC_0002.pdf">Passed</A>;
    %else %let VC_0002_status=<A HREF="VC_0002.pdf">Failed</A>;
%MEND check;
%check;
```

The form of this version of the %check macro is very similar to what was used in the enrollment numbers verification check example, with the only difference being that the logic test now checks whether the value of &numobs is 1 instead of comparing minimum and maximum values of num_repvar. It then follows the same process of including an HTML link with the “Passed” or “Failed” value of &VC_0002_status. &VC_0002_notes will become a string of the concatenation of all unique title1 statements, so in this case it will consist of the string “A Phase III Clinical Trial” since there was only one unique title1 for this report. VC_0006, which checks the title3 statements for consistency, uses the same procedure as detailed for VC_0002.

Title2 includes information regarding whether the TLF is for the Open or Closed version of the report, so an extra step is necessary to perform verification checks VC_0003 and VC_0004. In order to identify the data elements that are used in the Open report we use the INDEX function to subset to a data set consisting of only elements with a program ID that includes the word “open”. Similar code is used for VC_0004 to select Closed report elements. The program ID includes the directory path to the SAS® programs used for the report, with programs intended for the Open report stored under a path that includes the /open directory, and the same for Closed report programs.

```
DATA all_open;
    set save_all;
    if index(prgmid,'/open/') > 0;
run;
```

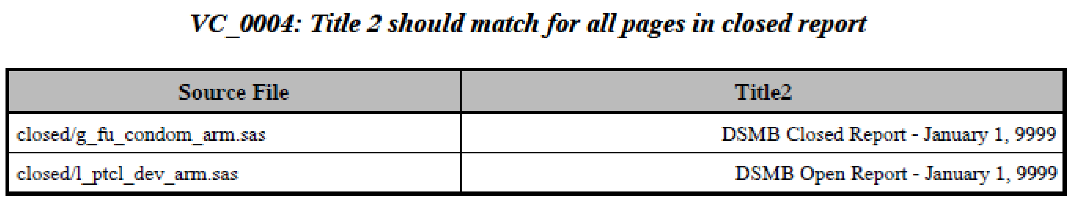
From this point on the process for completing the verification check is the same as it was for performing VC_0002. One potential drawback of this method is that if a program is stored under the wrong /closed or /open directory, or if it is stored in a different location that includes neither the word “open” or “closed” in the path, then the title2 statement for this program will not be included in the verification check. Display 9, Display 10, and Display 11 present the summary files for checks VC_0003, VC_0004, and VC_0005, respectively.



VC_0003: Title 2 should match for all pages in open report

Source File	Title2
open/g_accrual.sas	DSMB Open Report - January 1, 9999
open/t_retention_last3months_site.sas	Report Generation Date: March 25, 2015

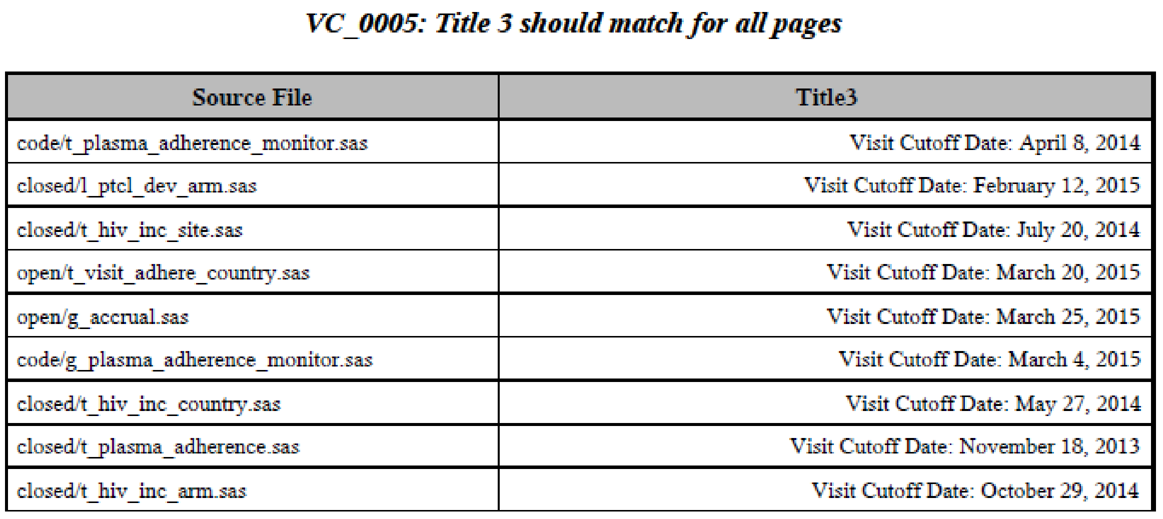
Display 9. Summary PDF for VC_0003



VC_0004: Title 2 should match for all pages in closed report

Source File	Title2
closed/g_fu_condom_arm.sas	DSMB Closed Report - January 1, 9999
closed/l_ptcl_dev_arm.sas	DSMB Open Report - January 1, 9999

Display 10. Summary PDF for VC_0004



VC_0005: Title 3 should match for all pages

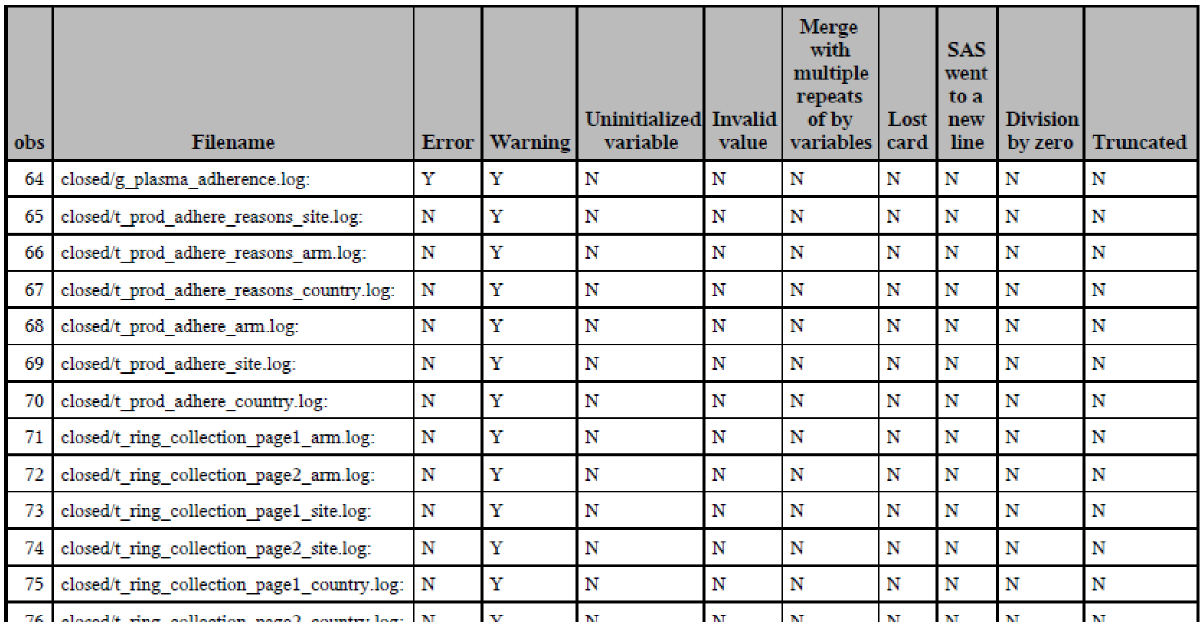
Source File	Title3
code/t_plasma_adherence_monitor.sas	Visit Cutoff Date: April 8, 2014
closed/l_ptcl_dev_arm.sas	Visit Cutoff Date: February 12, 2015
closed/t_hiv_inc_site.sas	Visit Cutoff Date: July 20, 2014
open/t_visit_adhere_country.sas	Visit Cutoff Date: March 20, 2015
open/g_accrual.sas	Visit Cutoff Date: March 25, 2015
code/g_plasma_adherence_monitor.sas	Visit Cutoff Date: March 4, 2015
closed/t_hiv_inc_country.sas	Visit Cutoff Date: May 27, 2014
closed/t_plasma_adherence.sas	Visit Cutoff Date: November 18, 2013
closed/t_hiv_inc_arm.sas	Visit Cutoff Date: October 29, 2014

Display 11. Summary PDF for VC_0005

The summary files show that these verification checks identified discrepancies in the title statements. For VC_0003 we see that at least one program has a title2 statement of “Report Generation Date: March 25, 2015”. For VC_0004 at least one program has a title2 statement that indicates the TLF is part of the Open report even though the SAS® program that generates that TLF is located under the /closed directory. Finally, VC_0005 shows that programs used many different cutoff dates, or at least that the title statements indicated that different cutoff dates were used. Since there are multiple observations in the data sets used for those summary files, all three of these checks should assign a value of “Failed” along with an appropriate HTML link to the %VC_0003_status through %VC_0005_status variables.

SAS® LOG FILE VERIFICATION CHECK

The final verification check that we will discuss is of a different nature than the logic checks described above. It is intended to illustrate how the verification check process can be used to check for a variety of potential issues. The programs that generate the TLFs for our SMC and DSMB reports run on a daily cron job, and as part of this cron job we create and store a summary of the log files for each program. The summary reports whether or not there were any occurrences of the phrases “errors”, “warnings”, “uninitialized variables”, “invalid values”, “merge with multiple repeats of by variables”, “lost card”, “SAS® went to a new line”, “division by zero”, and “truncated messages in program logs” in the SAS® logs of programs that ran as part of the cron job. If any of those key phrases did occur, the report will also contain the text from the line of the log in which they occurred. From this SAS® log file we identify occurrences of potential issues and determine whether each of these criteria have occurred for each SAS® program.



obs	Filename	Error	Warning	Uninitialized variable	Invalid value	Merge with multiple repeats of by variables	Lost card	SAS went to a new line	Division by zero	Truncated
64	closed/g_plasma_adherence.log:	Y	Y	N	N	N	N	N	N	N
65	closed/t_prod_adhere_reasons_site.log:	N	Y	N	N	N	N	N	N	N
66	closed/t_prod_adhere_reasons_arm.log:	N	Y	N	N	N	N	N	N	N
67	closed/t_prod_adhere_reasons_country.log:	N	Y	N	N	N	N	N	N	N
68	closed/t_prod_adhere_arm.log:	N	Y	N	N	N	N	N	N	N
69	closed/t_prod_adhere_site.log:	N	Y	N	N	N	N	N	N	N
70	closed/t_prod_adhere_country.log:	N	Y	N	N	N	N	N	N	N
71	closed/t_ring_collection_page1_arm.log:	N	Y	N	N	N	N	N	N	N
72	closed/t_ring_collection_page2_arm.log:	N	Y	N	N	N	N	N	N	N
73	closed/t_ring_collection_page1_site.log:	N	Y	N	N	N	N	N	N	N
74	closed/t_ring_collection_page2_site.log:	N	Y	N	N	N	N	N	N	N
75	closed/t_ring_collection_page1_country.log:	N	Y	N	N	N	N	N	N	N
76	closed/t_ring_collection_page2_country.log:	N	Y	N	N	N	N	N	N	N

Display 12. Summary PDF for VC_0006

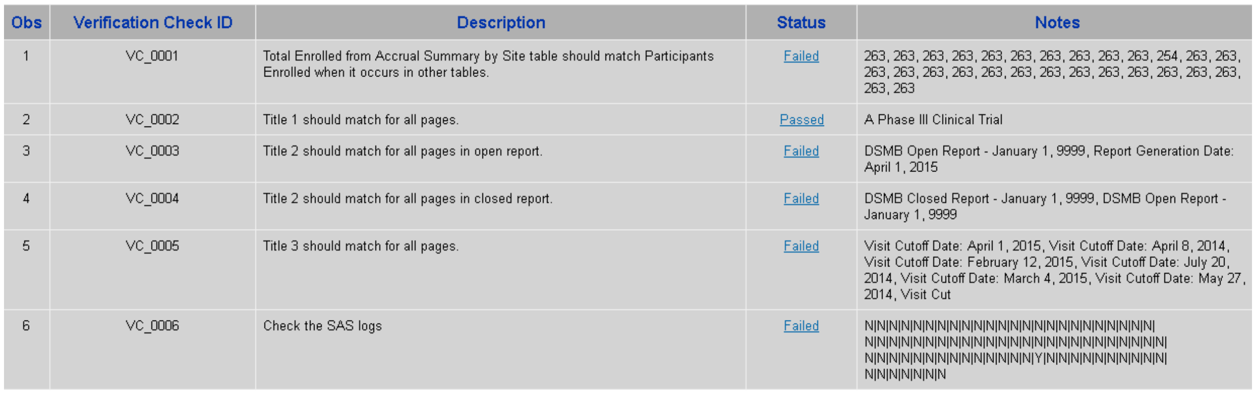
From the PDF summary in Display 12 we can easily see whether errors, warnings, or other potential issues have occurred for each program that was run as part of the cron job. We also use the slog data set as part of a verification check on the slog. We decided to consider any occurrences of an error message as a failure, but to allow the verification check to pass if warnings, uninitialized variables, or other potential issues occurred. In this example we observe that the g_plasma_adherence.sas program had an error, and so the verification check will fail.

UPDATE HTML OUTPUT RESULTS

Once all of the verification checks specified in the Excel Verification Driver file have run, we update our HTML template. With all of the verification check SAS® programs run we have macro variables &VC_<IDnumber>_status and &VC_<IDnumber>_notes stored for each check, and now need to add these to the HTML template that was created earlier. To do this we wrote a macro called %read_htm, as shown below.

```
%MACRO read_htm(webin,webout);
    filename web_in "&webin";
    filename web_out "&webout";
    DATA _null_;
           length line $ 256;
           infile web_in length=lvg;
           file web_out;
           input @1 line $varying200. lvg;
           line=tranwrd(line,'&amp;','&');
           line = trim(resolve(line));
           put line;
    run;
    filename web_in;
    filename web_out;
%MEND read_htm;

%read_htm(webin= Validation_template.htm,
          webout=Validation_dashboard.htm );
```

This macro reads the previously created HTML template file. It also takes advantage of the RESOLVE function, which substitutes &VC_<IDnumber>_status and &VC_<IDnumber>_status macro variables with the derived values. Once this macro has been executed we have built our verification dashboard, which is shown in Display 13.



Obs	Verification Check ID	Description	Status	Notes
1	VC_0001	Total Enrolled from Accrual Summary by Site table should match Participants Enrolled when it occurs in other tables.	Failed	263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 254, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263, 263
2	VC_0002	Title 1 should match for all pages.	Passed	A Phase III Clinical Trial
3	VC_0003	Title 2 should match for all pages in open report.	Failed	DSMB Open Report - January 1, 9999, Report Generation Date: April 1, 2015
4	VC_0004	Title 2 should match for all pages in closed report.	Failed	DSMB Closed Report - January 1, 9999, DSMB Open Report - January 1, 9999
5	VC_0005	Title 3 should match for all pages.	Failed	Visit Cutoff Date: April 1, 2015, Visit Cutoff Date: April 8, 2014, Visit Cutoff Date: February 12, 2015, Visit Cutoff Date: July 20, 2014, Visit Cutoff Date: March 4, 2015, Visit Cutoff Date: May 27, 2014, Visit Cut
6	VC_0006	Check the SAS logs	Failed	N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N| N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N|N| N|N|N|N|N|N|N|N|N|N|N|N|N|N|Y|N|N|N|N|N|N|N|N|N|N| N|N|N|N|N|N|N

Display 13. Verification Dashboard after Execution of Verification Checks

The Status and Notes columns have now been resolved, with Status reporting whether the check passed or failed and containing a link to the PDF summary file for that verification check. The Notes provide a brief description of the results of the check. VC_0001 failed because total enrollment did not match across all the checked TLFs, and we can see the enrollment totals from each of those TLFs in the Notes column. VC_0002 passed and, appropriately, the Notes column only contains one title statement. With VC_0003, VC_0004, and VC_0005 we can see that the checks failed under Status, and by looking at Notes we observe the discrepant title statements that are causing the failure. VC_0006 also failed, and by looking at the Notes column we can see that there was an error in the SAS® log of one program. Having a general idea of the causes of failure, we then follow the links to the summary PDF files in order to see which programs were causing the failures.

CONCLUSION

It should be noted that a limitation to our method is that verification checks are made on the SAS® data sets used to create the TLFs, rather than on the text contained in the PDF output of the TLFs themselves. If one of the TLFs in the report is created by a SAS® program that is not included in the checks for that report, or if the data set that creates that TLF is not part of the Results Data Set, then it is possible for discrepancies from that TLF to go unnoticed. Moreover, issues can also arise that we did not anticipate when writing the logic of the verification checks, and which subsequently are not caught by the automated process. Therefore, even when the checks identify failures, manual effort is still required to identify the source of those failures and to make the necessary corrections.

Overall, however, we found that developing an automated cross-report verification system using SAS® proved to be extremely helpful in completing the review process necessary to produce large and complex SMC and DSMB reports with high accuracy and integrity. By focusing on automating verification checks that were simple as well as those that would have been time consuming to do manually, we were able to focus our attention on the more complicated aspects of preparing the reports. Finally, we have incorporated the SAS® automated cross-report verification system into our daily crons and it has also proven to be useful in identify programming bugs and other problems early on in our daily reporting process.

CONTACT INFORMATION

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

Name: Daniel Szydlo, MS
Enterprise: Fred Hutchinson Cancer Research Center
Address: 1100 Fairview Ave N.
City, State ZIP: Seattle, WA 98109
E-mail: dszydlo@fhcrc.org

Name: Iraj Mohebalian, BS
Enterprise: Fred Hutchinson Cancer Research Center
Address: 1100 Fairview Ave N.
City, State ZIP: Seattle, WA 98109
E-mail: iraj@fhcrc.org

Name: Marla Husnik, MS
Enterprise: Fred Hutchinson Cancer Research Center
Address: 1100 Fairview Ave N.
City, State ZIP: Seattle, WA 98109
E-mail: marla@fhcrc.org