PharmaSUG 2015 Paper IB01

The 5 Most Important Clinical SAS® Programming Validation Steps

Brian C. Shilling, inVentiv Health Clinical, Cape Coral, FL

The validation of a SAS programmer's work is of the utmost importance in the pharmaceutical industry. Because the industry is governed by federal laws, SAS programmers are bound by a very strict set of rules and regulations. Reporting accuracy is crucial as these data represent people and their lives. This presentation will give the 5 most important concepts of SAS programming validation that can be instantly applied to everyday programming efforts.

Knowing the Regulations

There are several sets of rules and regulations that are required to be followed while analyzing and presenting clinical data. Knowing this group of regulations is very important, and at times a legal requirement.

HIPAA

The Health Insurance Portability and Accountability Act was put into place in 1996 to provide rights and protection for participants and beneficiaries in group health plans. HIPAA has little to no impact on your day-to-day work as a programmer but it is important to understand that the law exists and to have a general idea of its purpose. In simple terms, HIPAA serves to protect the information about a subject's identifying information.

The Code of Federal Regulations

Title 21 of the Code of Federal Regulations, or CFR, pertains to food and drugs. Chapter 1 pertains to those components and identifies the Food and Drug Administration (FDA) and the Department of Health and Human Services (DHHS). Part 11 of this regulation is what pertains to you as a programmer. This chapter specifically identifies electronic records and electronic signatures. There are numerous topics within Title 21 that directly (Part 11 and Part 820) or indirectly (Part 50) affect programming. While you don't need to reach each of these, it is helpful to understand what parts of the clinical trial and programming process are driven by these rules.

International Conference on Harmonisation of Technical Requirements

In a global setting, it is important for all international parties involved in the drug development process to follow a standard set of definitions for similar concepts and a common understanding for how drugs should be developed, The International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH) is a global organization that provides these common definitions and guidelines and is often the source for standard values for certain data. Again, these requirements may not impact your programming responsibilities directly but they are part of the framework that built the studies and the specifications you work with regularly.

What is Validation

Validation is defined as "an act, process or instance of determining the degree of well-groundedness or justifiability: being at once relevant and meaningful.…"[1] Validation, a very in-depth process, is the act of proving that the outcome of a program accurately and effectively represents the original source clinical study data. This takes on many forms and involves a number of procedures. It is a *justification* of the means used to accomplish the outcome of the program and its accurate representation of the original data. This justification requires more effort and granular detail to prove that the data manipulation and analyses were performed correctly and appropriately.

Presenting Correct Information

Important decisions on subject health care are made based on the output generated by programmers. If incorrect information is presented as fact, people's lives could be at risk. It is essential that all data from a clinical trial are presented as accurately as possible so that decisions can be made appropriately. While there is constant pressure in the industry to produce output faster, if that output is incorrect, it could be worse than useless – it could actually be harmful. Careful, diligent validation is the best way to detect and prevent errors

Having a Plan

It is critical to have a plan before you begin any task in programming. Often it is as simple as being clear about the task you need to perform (e.g. create a summary table), gathering all of the sources of information you'll need to perform the task. Unless timelines prohibit it, programming the data listing *before* programming the table gives you one more chance to look at the data. This helps not only to validate the analysis data set but the original data that went into it as well.

Make the Code Do the Work

One of the best ways to be efficient is to make the program itself do as much of the checking for you as possible. If the calculations in your program assume that data exists or that it is consistent, programming "data traps" in your code to print any cases that do not meet your assumptions is often helpful.

Methods

There are generally two ways to approach validation of output: separate, independent programming and peer review. Each method has it strengths and weaknesses and there is no one correct way.

Documentation

This validation list outlines the basic requirements with which to begin. This list will grow as different tasks and types of data are discussed.

- Clean Logs
- Adequate, accurate notations
- Output looks like the specification
- Output content matches the specification

- Output content matches similar content on other output
- Output content makes sense

Validation Techniques

The following are several techniques that can be employed to aid in the process of validating the programming used in clinical trial data reporting:

Procedures

There is a wide variety of procedures provided within SAS that help do everything from sorting data to generating complex statistical analysis. While most people tend to think of the more complex procedures and how to use them, it is the simple procedures that are most helpful for validation of data and logic. PROC PRINT, PROC MEANS, PROC FREQ.

SAS Options and Language Elements

Use the SAS options to your advantage: MPRINT, SYMBOLGEN, and MLOGIC

Macros

The general rule for truly efficient programming is to use macros only when they add significantly to the process. Simple code that is used repeatedly throughout the many programs makes it more appropriate for macro usage.

Logs

One of the most important and simplest steps to making validation easier is to start with a clean log. This means that your log should not only be free of errors, but it should be free of warnings and "helpful" notes

Keeping/Dropping data

By keeping only the variables that you need, you will cut down on run times as well as creates a smaller, more manageable set of data.

Familiarity with Data

There are many types of data that are common across clinical trials. These data types usually consist of subject characteristics and indications of the subject's general state of health at a given time. When validating programs that reference clinical trial data, it is important to understand what makes sense for each data type to ensure that the methods used to validate the program are appropriate and that the output created makes sense. Even the most detailed specifications are no substitute for understanding what potential data issues to look out for and how data interrelate. This is the knowledge that employers are looking for when asking for pharmaceutical experience – they know that an in-depth understanding of the data is the critical element to effective programming and validation in the pharmaceutical industry.

In the process of creating analyses and summary reports in support of a CSR, analysis data sets are often created first to facilitate the creation of those reports. Once those

analysis data sets are created, there are often additional data manipulations and summary statistics generated in the process of creating the final TLF output. Throughout all of these tasks, the data being worked with needs to be considered in both the programming and validation process. Different types of data are expected to have different structures and the content is expected to behave in different ways. Regardless of whether you are manipulating data to create a permanently stored analysis data set or manipulating data just to create a report, it is important to understand the data so you can validate the result accurately and completely.

Reporting and Statistics

For all of the seeming variation in the data collected and the methods of collection in clinical trials, the reports generated on much of these data are surprisingly similar. Subject demographics, relevant medical history, physical exam findings, laboratory test results, and many others are usually reported using the same summary statistics. In many cases, the programming involves reporting the summary statistics from PROC FREQ and/or PROC MEANS (PROC UNIVARIATE). The techniques for summarizing the data and putting it together for the report are the same regardless of the output file type (text files, RTF, PDF, etc.) or layout (portrait or landscape, data in rows vs. columns, etc.). The validation of these summary reports and data listings often constitutes the bulk of the validation effort done for a project. In essence these analyses are the final product that will be added to the Clinical Trial Report (CTR) and used to make statements and conclusions about the safety and efficacy of the drug or device being studied.

As mentioned earlier, there are generally two approaches to validation – independent programming and peer review. Before either of these approaches comes into play, the programmer who is responsible for generating the final, "production" output must validate his or her own work. Regardless of the data being analyzed or reported and regardless of whether you are responsible for the production output or for validating someone else's output, there are general principals that apply to validating different categories of output (tables, listings, figures).

Pre-Output Validation Steps

One of the key elements of the validation process is the review of SAS code and SAS logs. No matter what type of output you are creating or the data with which you create it, the code itself needs to make sense and the log needs to be free of errors and warnings. While the final output is the ultimate product being validated, starting validation with the code and the log will increase the probability that the final product will be accurate and correct.

Code Review

It is critical that code be easy to read and contain enough comments to allow easy understanding of what is being done (and sometimes why). It is good practice to review your own code after it is written to make sure that the comments make sense and are sufficient to explain what is being done. Regardless of whether you are reviewing your own code or your peer's as part of the validation process, code review (reviewing the .SAS file) is an important step and the final code should meet the following criteria:

1) Is the code readable and understandable?
2) Are there sufficient comments such that another programmer could read and understand what is happening?
3) Is there a logical and reasonable flow to the program?
4) Does the code make sense in relation to the specifications? Is it reasonable to assume that the final outcome being created from this code would be accurate?
5) Are there any logic flaws or weaknesses where the code could fail?
6) Does the code adhere to the company standards?

It is important that the code be reviewed with these questions in mind. If the code itself is able to pass these criteria favorably, it will be less likely to have major issues with the final output and any minor issues will be much easier to trace.

Log Review

Log review is another important step in the validation process. Any messages of concern can be cause for speculation around the accuracy of the final output. Each and every SAS log should be scanned and reviewed for SAS notes, warnings and errors. You might want to consider looking for, at the very least, messages that start with the following keywords:

ERROR
WARNING
INFO: Character
INFO: The variable
NOTE: At least
NOTE: Character
NOTE: Division
NOTE: Mathematical
NOTE: Merge
NOTE: Missing
NOTE: NOSPOOL
NOTE: Numeric
NOTE: Variable

While the warning and error messages are clear problems, the notes listed above are often more subtle indications that the data or the code is not behaving as expected. While SAS is able to continue processing the data, it may not be doing what you really intended. It is critical to understand what each of these notes means and ensure that the results are what you intended. Once you know the source of the note, even if SAS handles the data correctly, it is recommended that you adjust your code such that these notes do not appear. This way, if code needs to be run in the future these notes will not be cause for concern.

In addition to simply checking for notes and warnings, it is important to follow the number of observations from one step to another. It is possible for code to execute with no notes or warnings, but due to issues in the data or unexpected problems with the code logic, the final result is unexpected. In many cases this will be evinced in the number of observations being different than expected. Prior to reviewing the final output, it is

critical to review the log to make sure the number of observations flowing into and out of each data step or procedure makes logical sense. Chances are that if the number of observations doesn't make sense, the final output won't make sense either.

Conclusion

The clear and accurate representation of clinical trial data is crucial. Careful and complete validation of clinical datasets, summary statistics and other reports generated by programmers is absolutely critical in proving that the results are accurate. Using these methods to validate various types of data and output will enable you to confidently deliver a validated, high-quality product that accurately represents the clinical study data.

References

[1] www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=valid(Merriam-Webster's Online Dictionary)

Acknowledgements

I would like to thank my employer inVentiv Health Clinical for allowing me the time to develop and present this paper.

Contact Information

Your comments and questions are valued and encouraged. Please feel free to contact the author at:

Brian C. Shilling
inVentiv Health Clincal
533 SE 21st Avenue
Cape Coral, FL 33990
Email: bcshilling@gmail.com