PharmaSUG 2013 - Paper TF23

Effective Independent Validation

Tips to Improve the Independent Validation Process

Daniel Butner, PPD, Wilmington, NC
Brandon Graham, PPD, Wilmington, NC

ABSTRACT

Independent validation, a technique commonly used in the clinical research industry, is the act of two programmers working independently from a common set of specifications and comparing their results. This process enhances the quality of both the specifications and the output since the two programmers must reach the same result using independent approaches before it is considered “validated”. Skilled SAS® programmers who are effective at independent validation can greatly increase the quality and efficiency of a project.

This paper examines the technical approaches that may be used to identify output differences between independent programmers. It includes approaches to validating data sets as well as “standard output” (e.g. table, list, or graph output). In addition, specific techniques to improve efficiency and detail are discussed.

This paper also gives guidance on providing validation feedback. It discusses considerations with regard to the amount of detail provided as well as methods for providing feedback, and offers tips for communicating effectively during the validation process.

INTRODUCTION

Though independent validation is used commonly in the clinical research industry, it has been our observation that methods of independent validation can vary from organization to organization. Further, our experience has shown some SAS programmers seem to be naturally (or by training) more effective than others at providing validation feedback in a manner that is conducive to quick resolution, and in a way that enhances collaboration and promotes healthy co-worker interaction.

This paper will examine some key concepts of independent validation and provide some ideas which we believe can make the process of independent validation more efficient and may also make the process a more pleasant experience for those involved.

SO, WHAT EXACTLY IS “INDEPENDENT VALIDATION”?

While we recognize there may be subtle differences in the processes used throughout the industry, for the purposes of this paper we will define independent validation as follows: two programmers working independently yet collaboratively to come to the same result in order to verify that their output is accurate.

We will assume that each “independent” programmer works from the same specification, writes and maintains their own code, and does not share any code or specifics of the approach used to come to a result. However, we will assume that the independent programmers can collaborate with each other and provide examples which reference the source data to explain how they came to a result.

The specification can be authored by a third party, like a statistician, or by one of the two programmers. Before programming starts the specification should be reviewed in detail by both programmers. It is assumed that some small changes to the specification may be needed during the validation process. It should be noted that working from specifications that are not final may impact the quality and will almost certainly reduce the efficiency of the independent validation process. The process described in this paper assumes that the specification has been reviewed by both programmers and is considered final.

APPROACH TO IDENTIFY DIFFERENCES

There is a great deal of SAS documentation and many conference papers which discuss the technical details of PROC COMPARE and its options which can be used to assist in determining differences between two SAS data sets. While specific SAS procedures and options are beyond the scope of this paper, we would like to present a basic process which we have found to be useful in investigating differences between programmer and validator, particularly to programmers who are new to the process of independent validation.

DATA SET VALIDATION

Suppose we are validating a transformed or derived data set and our PROC COMPARE result does not match. What do we do? A basic process flow can be seen in the decision diagram below (Figure 1). See references to Steps 1 – 3 below for specific tips and tricks that may be useful at each level.

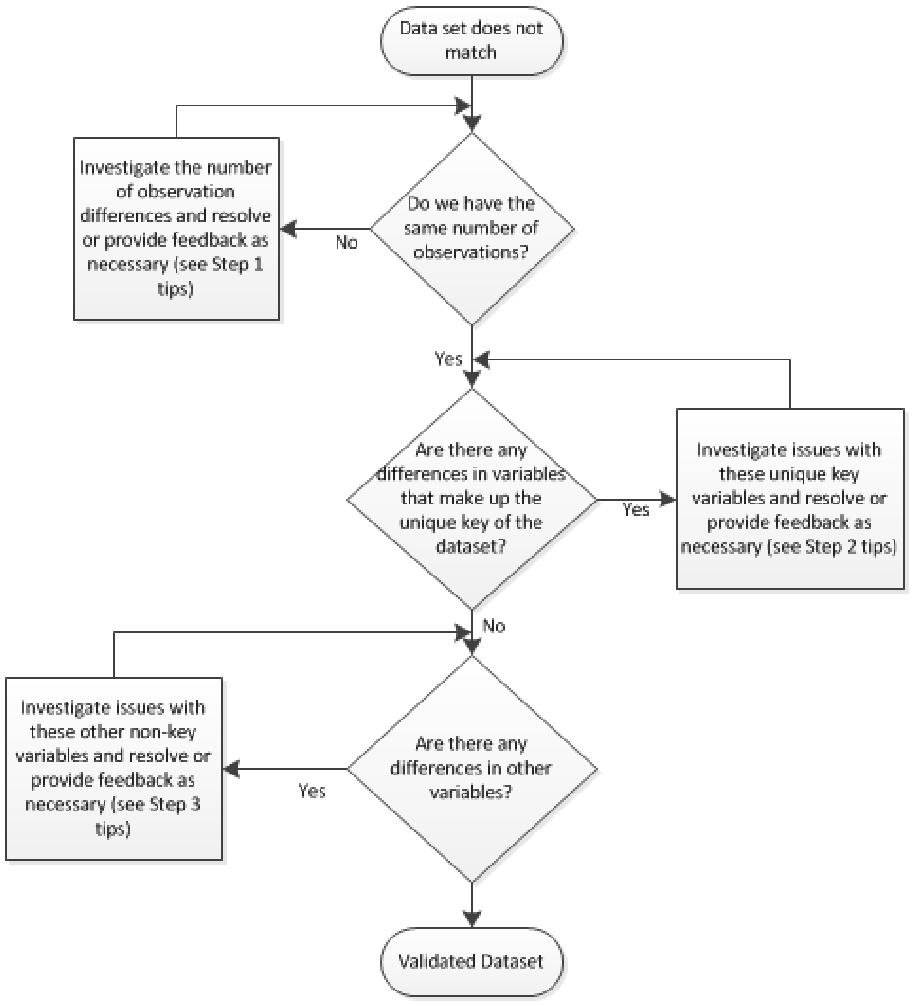
Data set does not match
Investigate the number of observation differences and resolve or provide feedback as necessary (see Step 1 tips)
No
Do we have the same number of observations?
Yes
Are there any differences in variables that make up the unique key of the dataset?
Yes
Investigate issues with these unique key variables and resolve or provide feedback as necessary (see Step 2 tips)
No
Investigate issues with these other non-key variables and resolve or provide feedback as necessary (see Step 3 tips)
Yes
Are there any differences in other variables?
Validated Dataset

Figure 1. Data Set Validation Process

Step 1 (differences in number of observations) tips:

- Check your SAS log. Examine the number of observations after each data manipulation step to ensure the validation has the correct number of observations based on the source data and specification.
- If you determine the data set being validated has an incorrect number of observations, feed this information back to the programming counterpart, providing as much information as possible about the differences in number of observations you observe.

Step 2 (differences in unique key variables) tips:

- Ensure that the sort order is consistent between the data sets being compared.
- Ensure the unique key variables showing differences don't have any spelling, concatenation, or truncation issues.
- Confirm that your key variables truly identify unique records and that the data is sorted by these variables.
- When programming and validating on "dirty" or incomplete data, a lack of data in the unique key variables

can often cause differences in the sort order between programmer and validator. Ensure that you truly have a unique key based on the data in the source data set. Adding additional variables to the sort order may be needed when key variables are incomplete.

Step 3 (differences in non-unique key variables) tips:

- Ensure that the key variables are validated first. Often if the key variables are not matching (and the issue is due to sort issues) it is difficult or even fruitless to compare non-key variables.
- Take it one variable at a time. Look at both programmer and validator data for each variable that differs. It may be helpful to subset data based on specific data values (such as patient identifier).
- Check the values in the original source data against the specification and your code to ensure there are no spelling, concatenation, or truncation issues.

At all points in this process, remember to refer back to the specification. The specification is the road map and documentation for both the creation and validation of the data set. If additional clarification is needed in the specification, ensure that sufficient detail is added in a timely fashion throughout the process. It is very important for the program and the programming specification to remain in sync.

STANDARD OUTPUT VALIDATION PROCESS

Before getting into the details of table, listing, and graph validation, we would like to take a look at the validation process for standard output at a high level. The two process diagrams below detail validation processes that are common within the industry. These two validation processes are similar, but differ in the source used for comparison in the validation. In Process A, one programmer creates a SAS data set which is used for comparison by the validation counterpart. Process B involves parsing the final output directly for comparison by the validation programmer.

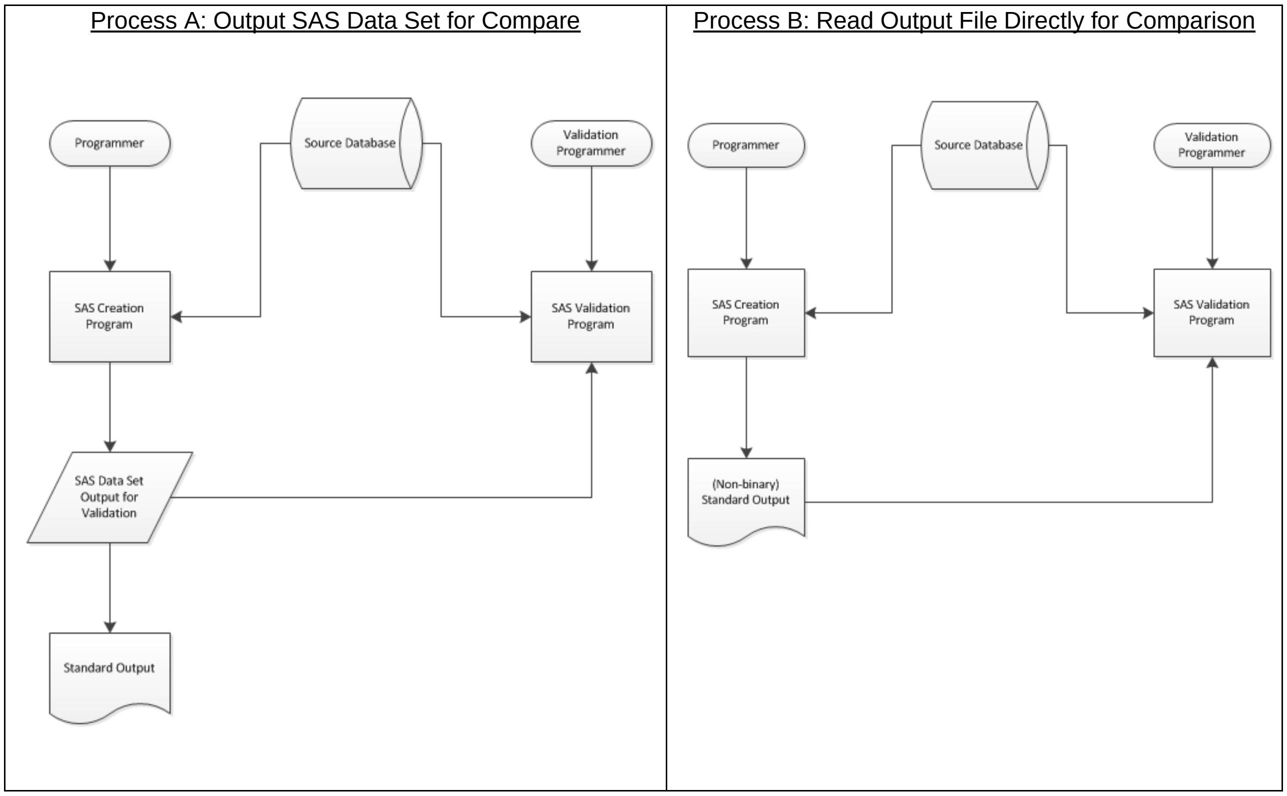

Figure 2. Table/Listing Validation Processes

Depending on the output type, Process B is not always possible. Graphical output types are binary and force programmers to use Process A. However, common table and listing output types such as RTF and ASCII can be parsed to extract the output data directly from the table or listing output and into a data set for validation comparison, making Process B an option. We advocate parsing the table or listing output directly (Process B) whenever possible. Parsing the table and listing output file overcomes the main weakness with Process A: that data manipulation is possible in the final output step (most commonly PROC REPORT) and these data manipulations are not being fully validated through the data set comparison. Validating the final output file directly does require some additional effort but this extra effort helps to more fully close the validation loop and should result in higher quality output.

Considerations for Table, Listing, and Graph Validation

Regardless of the validation process you or your organization choses to implement, there are some specific validation strategies at the individual table/listing/graph level which will improve the overall quality of output validation. The following are a couple of key concepts we think improve the quality of a validation effort.

1.) Compare as much as you can programmatically

While human beings are good at a lot of tasks, they cannot compete with computers when it comes to tedious and repetitive tasks. With that in mind, we want to set up the validation so that as much of the output as possible is compared programmatically.

For table and listing output, this includes comparing not just the data in the body of the document, but also any row and column header information and any calculated values included in the titles or footnotes whenever possible. This will take a little additional time to set up initially, but this effort will be well worth the time when multiple reruns are expected.

For graphs, ensure that the entire data set which is being input into the graphing procedure is being validated. In addition, if the annotate facility is being used to write data or add enhancements to the graph, we recommend that the annotate data set be programmatically compared as well.

2.) Take advantage of the SYSINFO code from PROC COMPARE

The automatic SYSINFO macro variable is updated each time a PROC COMPARE is executed. The value of the SYSINFO variable returned represents the results of the comparison. Taking advantage of the information returned via this SYSINFO variable can be an excellent way to produce a summary of validation results. See the PharmaSUG paper by Lex Fennell listed in the recommended reading section below for detailed information on how this can be accomplished. This method can also be applied to data set validation.

PROVIDING GOOD VALIDATION FEEDBACK

METHODS OF PROVIDING FEEDBACK

Once you have identified a difference in the validation results, it is time to consider the best method to communicate this information. Although there is no single "correct" method, you should consider the complexity of the issue being communicated, the urgency of receiving a quick response, and any communication preferences which have been discussed with your counterpart. It may be worthwhile to establish a preferred method in advance if you are working with someone who has a strong preference for or against a specific method. We have listed some common methods of communication and some pros and cons of each in Table 1 below.

Method	Pros	Cons
Email	• Easily record conversation for future reference • Not as intrusive as other methods	• Can be slower than other methods • Lack of tonal reference • May lead to more misunderstandings or misinterpretations
Instant Message (IM)	• May lead to quick response • Conversation can be saved for future reference	• Can be intrusive or distracting • Lack of tonal reference (although many IM services offer quick-pick emoticons which helps to replace tonal reference)
Verbal (via phone or in person)	• Often leads to quick response • Some issues may be easier to discuss verbally than to explain in a written message • Ability to add tonal reference lessens risk of misinterpretation	• Most intrusive or distracting method • No written record for future reference

Table 1. Pros and Cons of Communication Methods

COMMUNICATION STYLE

The ability to clearly present and request information, to express yourself verbally, to read nonverbal communication, to actively listen, to explain complex concepts, and to understand cultural differences in communication in a global workforce are great assets. Different people also have different personality types and communication styles.

Therefore the style of your communication is also important to consider. We believe effective communication in the independent validation process contains two key elements:

1.) It should be detailed and clear enough that your coworker can easily understand the validation issue that you have uncovered and why you think the current output is not correct

2.) It should be communicated in a tactful and professional manner that promotes a healthy and collaborative work environment

Let's take a look at a simple example below of a coworker providing feedback during the validation process of a DM data set.

Example Communication	**Detailed**	**Tactful / Professional**
"Your DM data set is wrong, please fix it."	No	No
"I believe your DM data set is incorrect. Please take a look and let me know if you disagree. Thanks."	No	Yes
"There are only 10 patients in our study; DM is a patient-level data set and should have 1 record per patient. Your output is wrong, fix and let me know."	Yes	No
"Currently I see 12 records in your DM data set. I was expecting to see only 10 records since DM should be a patient-level data set. Please take a look when you have a minute and let me know if you disagree. Thanks."	Yes	Yes

Table 1. Communication Examples

As you can see from the examples above, taking just a little more time to ensure that your validation feedback is detailed enough to explain the issue can save the coworker time when he or she is investigating and resolving the issue. More complex issues typically require some time to provide a detailed explanation, but overall the validation process is more efficient if both sides gain a clear understanding of the issue through detailed, tactful communication. *Important Note: We are only discussing sufficient detail to explain the difference in results, not sharing code or discussing specifics of the techniques used.*

CONCLUSION

Independent validation can be a very effective method to ensure high quality results if one follows a few basic rules: do your own research to attempt to identify the problem, provide detailed information, and be mindful of both your own communication style preferences and that of your colleagues. Programmers who excel at this process can improve efficiency within any organization. We hope that you have come away with a better understanding of why independent validation is used, as well as some practical considerations that can be implemented to improve the validation process.

REFERENCES

La Brec, Paul A. 2010. "The Managers Skill Set: How to Move from Programmer to Manager". Proceedings of the 2010 PharmaSUG Conference. Available at http://www.lexjansen.com/pharmasug/2010/ma/ma06.pdf

Fennell, Lex. 2006. "An Easy, Concise Way to Summarize Multiple PROC COMPAREs Using the SYSINFO Macro Variable". Proceeding of the 2006 PharmaSUG Conference. Available at http://www.lexjansen.com/pharmasug/2006/TechnicalTechniques/TT23.pdf

ACKNOWLEDGMENTS

We would like to thank our PPD colleagues, particularly Tammy Jackson, Tom Fritchey, and Rob Diseker who provided suggestions which improved this paper.

RECOMMENDED READING

- "An Easy, Concise Way to Summarize Multiple Proc COMPAREs using the SYSINFO Macro", Lexter Fennell: http://www.lexjansen.com/pharmasug/2006/TechnicalTechniques/TT23.pdf

CONTACT INFORMATION

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

Daniel Butner
PPD
929 N. Front St.
Wilmington, NC 28401
Work Phone: +1 (336) 293-7259
E-mail: daniel.butner@ppdi.com

Brandon Graham
PPD
929 N. Front St.
Wilmington, NC 28401
Work Phone: +1 (910) 558-7386
E-mail: brandon.graham@ppdi.com