PharmaSUG 2016 - Paper QT11

Data Validation: Bolstering Quality and Efficiency

Anusuiya Ghanghas Novartis Healthcare Private Limited, Hyderabad, India
Houde Zhang, Novartis Pharmaceuticals, East Hanover, NJ, United States of America
Rajinder Kumar, Novartis Healthcare Private Limited, Hyderabad, India

ABSTRACT

We are working in health care industry and here a small mistake means risk for many lives. Statistical programming is a small area of this industry but our tiny efforts help people to live a better life. It is necessary that all the work done here should be done with higher quality. To maintain quality, validation process needs to be defined. This paper includes general outline of two validation processes of statistical programs used in clinical research. In the first process, validation programmer will not write independent code but manually check results or small portion of results. This is normally preferred in analyzing small data. In the second process, validation programmers will write their own code and compare the final datasets with programming codes or procedures (PROC compare). The second process is more precise, less dependent and more aligned. This paper also includes efficient way of second validation process which generally takes lot of time and that is sometimes more painful than creating original results. For that, initial programmer will save their final dataset in predefined format that will be same for all the programs based on the type of report.

INTRODUCTION

In today's fast life everyone wants to get things done quickly, but when it comes to validation, sole purpose of which is to ensure quality, we want it to be done in speedy way, with higher quality. This paper puts light on basic concept of validation, need and purpose of validation and then with the help of tips, how to perform it in a better way. It also provides insight of different validation techniques/plans used in general. It helps programmers to perform validation in efficient way by saving some time.

1.0 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". Generally work done by validator helps to ensure quality in first place and its importance can't be ignored. After knowing what validation is, one would be interested in knowing why there is need for it? Programmers being human being, it is quite possible for having some minor mistakes or ignorance, which can be due to time constraints hence it becomes necessary to have validation process defined and followed for ensuring desired quality.

2.0 TYPES OF VALIDATION

As per the complexity of outputs, type of validation changes, it also depends on many other things like prior experience (being standard output or not), criticality level, size of output and complexity of derivation used in generating the outputs etc.; based on all these mainly validation type can be divided into below categories.

2.1 DOUBLE PROGRAMMING WITH MANUAL CHECK

Validation programmer writes its own code for creating separate output and checks values manually. This type comes into picture when some derivation is used in generating the output and in general output size is small. Especially this is good for tables having output of 1 or 2 pages. Its less efficient, time consuming and not 100% full proof for bigger outputs.

2.2 DOUBLE PROGRAMMING WITH PROC COMPARE

Here validation programmer writes its own code to derive validation dataset or outputs then compare them against the ones created by production programmer. This is good for complex and critical outputs, where one want to ensure to check everything. It helps to validate accurately and efficiently, but drawback is that to know each bit of information in terms of name, length and format of variables. Going forward will discuss further later in this paper.

2.3 FORMAT CHECK OF OUTPUTS

Being it same type of trials and same type of outputs, there are few outputs which are quite standard, means they are created in same way and mostly with the help of existing program. Hence validator programmer needs not to write

program and check entire thing. In such cases only checking formatting of outputs is sufficient. General demographics tables can be such examples.

2.4 CONTENT CHECK OF OUTPUT

This is performed as part of senior review or by statistician for checking if content of output is correct or not, where derivation and logic are checked and counts and other numbers are confirmed in output.

2.5 REVIEW SAS CODE AND SAS LOG

In some cases validator prefer to check SAS log of production programmer for checking subset criteria and logic applied for deriving something and also checks SAS log for ensuring no error, unwanted warning or note is present in log. If someone don't want to open and check the log of production programmer then there is a macro code available for checking log for errors, warnings and notes, which can be used for getting the summary of these unwanted message simply by passing the location of log file and without going to particular location and opening it. Code for this log check macro can be easily found at www.support.sas.com website.

2.6 CROSS CHECKING

While validating, programmer checks output's correctness with respect to specification and data. For consistency purpose validation programmer also checks in different outputs, if same thing is displayed then it should be same in all the output. This type of check is called cross checking. For example overall AE table counts can be cross checked with individual AE tables.

3.0 VALIDATION APPROACH

Entire approach for validation (with double programming with PROC COMPARE) can be divided into three major parts:

- Datasets
- Tables, figures and Listings(TFLs)
- PROC COMPARE

3.1 DATASETS

Validation of datasets(analysis datasets/SDTM/ADaM) is bit simpler for validation programmer as metadata is fixed in this case, which means name and label of dataset, name and number of variables along with their attributes such as type, label, length, format, informat etc. are predefined. Being it simple and process driven where all major parameters are predefined, hence very minimal co-ordination between both (validation and production) programmers is needed.

3.2 TABLES, FIGURES AND LISTINGS (TFLS)

Validation of TFLs is bit tough for programmer as metadata is not fixed here and as a result better co-ordination is needed between both the programmers. For overcoming from this non-fixed metadata issue two simple solutions are available as below:

- Either both programmers should use pre-defined approach of naming convention and their type (character/numeric) for variables.
- Or validation programmer should keep same name and type of variable as followed by production programmer, or it should change name and type of variables of production programmer as per its own approach.

First approach of following pre-defined naming convention and type is better one. Usually it is recommended for using variables names as C1, C2, … etc. as per column number in output and keeping all of them as character only.

3.3 PROC COMPARE

PROC COMPARE would be needed for both datasets and TFLs, with some basic feature of PROC COMPARE procedure it becomes really easy to debug the mismatches or compare the result. For better use of PROC compare procedure following paper can help in a better a way.

http://www.pharmasug.org/proceedings/2015/IB/PharmaSUG-2015-IB11.pdf

Sometimes a simple line at the end of PROC Compare result as "`NOTE: No unequal values were found. All values compared are exactly equal.`" can be misleading also. First of all validation programmer

should check name and label of both the datasets which are getting compared, which helps in ensuring right datasets are picked for comparison purpose. Then by checking date of last modified for output, it helps in checking if latest dataset is taken. Then total number of variables and total number of observations should be compared, if it is equal in both the datasets then its good otherwise it should be checked which one is correct. Even having same number of variables/observations in both the datasets doesn't guarantee for its correctness, because validation programmer would be more interested in no. of common variables/observations. If above all things are fine then values of all the variables for individual observations are checked. If it also matches then it gives immense pleasure and relief to validation programmer.

CONCLUSION

This paper tries to define validation, types of validation and different approaches used in general for validation. Validation being an important aspect of ensuring quality, it can be good help for performing validation efficiently and in saving some time.

REFERENCES

- www.support.sas.com
- http://www.lexjansen.com/pharmasug/
- Carol I. Matthews and Brian C. Shilling. 2008. Validating Clinical Trial Data Reporting with SAS. Cary, NC: SAS Institute

ACKNOWLEDGMENTS

The authors take this opportunity to thank Muralikrishna Chakravarthula manager-lead statistical analyst at Novartis Healthcare Private Limited and Sudarshan Reddy, Senior Statistical Programmer at inVentiv Health Clinical, whose support and guidance encouraged us to write this paper.

DISCLAIMER

The contents of this paper are the work of the authors and do not necessarily represent the opinions, recommendations or practices of Novartis Healthcare Private Limited.

CONTACT INFORMATION

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

Name: Anusuiya Ghanghas
Enterprise: Novartis Healthcare Private Limited
City, Country ZIP: Hyderabad, India -500032
Phone: +91 - 8421484652
E-mail: this.anusuiya@gmail.com

Name: Houde Zhang
Enterprise: Novartis Pharmaceuticals
City, Country ZIP: East Hanover, NJ, USA
E-mail: onmyway2007@yahoo.com

Name: Rajinder Kumar
Enterprise: Novartis Healthcare Private Limited
City, Country ZIP: Hyderabad, India -500032
Phone: +91 - 9545988828
E-mail: rajinder_sihag@yahoo.co.in