

Using R to Validate Results of Other Programming Environments

Nikhil Abhyankar, Cytel, Pune, India
Vidyagouri Prayag, Cytel, Pune, India

ABSTRACT

During the course of statistical software development, validation of algorithms is a crucial activity which ascertains the accuracy and quality of the numbers they produce. The open source R software can be used extensively in achieving this goal. The set of validation tests provided by the Comprehensive R Archive Network (CRAN)¹ and the growing community of R enthusiast statisticians and computer scientists make it possible to use R for this purpose. The open source nature and the various contributed packages facilitate its use for testing advanced theories, especially when there are no other products available for comparison of first-to-market products. In the clinical data analysis industry, SAS® is the preferred software for statistical analyses. However, R can be used for independent and reliable validation of outputs ranging from subject listings to advanced analyses. This paper briefly describes the process of validation of software under development and of analysis carried out using other software in clinical data analysis.

INTRODUCTION

Submission of statistical analysis of clinical trials data to FDA is a crucial step during drug development. FDA needs to be assured that the results are accurate, consistent and reliable. Statistical software plays a major role in the clinical trials domain. Specialized software helps in the design, simulation, monitoring and analysis of data in a regulated trial. Validation of software or results of statistical analyses is necessary to ensure this. The FDA, concerned with the safety of the participants in clinical trials, has issued guidelines on software validation, electronic records maintenance and the use of computerized systems in clinical trials.

FDA GUIDELINES AND REGULATIONS FOR VALIDATION

In the guidance document, 'General Principles of Software Validation; Final Guidance for Industry and FDA Staff'², FDA defines validation as, "Establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specifications and quality attributes" The document goes further into software validation. The salient points are:

- Established software requirements specification is useful as a baseline for validation
- Use of a mixture of methods and techniques to prevent and detect software errors is recommended
- Validation to be conducted whenever there is a change in the analysis or a software product
- Independent evaluation is essential

R AS A TOOL FOR VALIDATION

The R software can be used for the validation of both, statistical software and analyses. The open source software along with the numerous packages enables the user to explore data in an effective and rapid manner. An R package is a directory system containing R-code, data, documentation, and compiled C++ .dll files that can all be loaded together (eventually) with a single `library()` command.

The kind of data acrobatics offered by R coupled with the availability of latest theories make it an ideal choice for validation. R can be used to validate the results of analyses ranging from the routine safety summaries to more advanced techniques like mixed models. Validation carried out in software other than the one used for analysis is more reliable as the default assumptions could be different in both. Also, for any deviations from the planned analyses, the R validator can be appropriately and conveniently modified to provide validation.

On the software development side, R can be used right from intermediate internal checks to creating and executing test cases for the quality assurance function. For the first to market products which incorporate an innovative theory, R is one of the preferred choices for writing validation codes in the absence of other products to compare with.

USE OF R AT CYTEL

QA IN PRODUCT DEVELOPMENT

Cytel develops industry leading software like *StatXact*[®], *East*[®], *SiZ*[®], *Compass*[®] and others. These are validated against similar established products, relevant R packages and in-house R codes developed independently for the purpose. The basic flow of the process of validation of algorithms developed is depicted in Figure 1 followed by its detailed description.

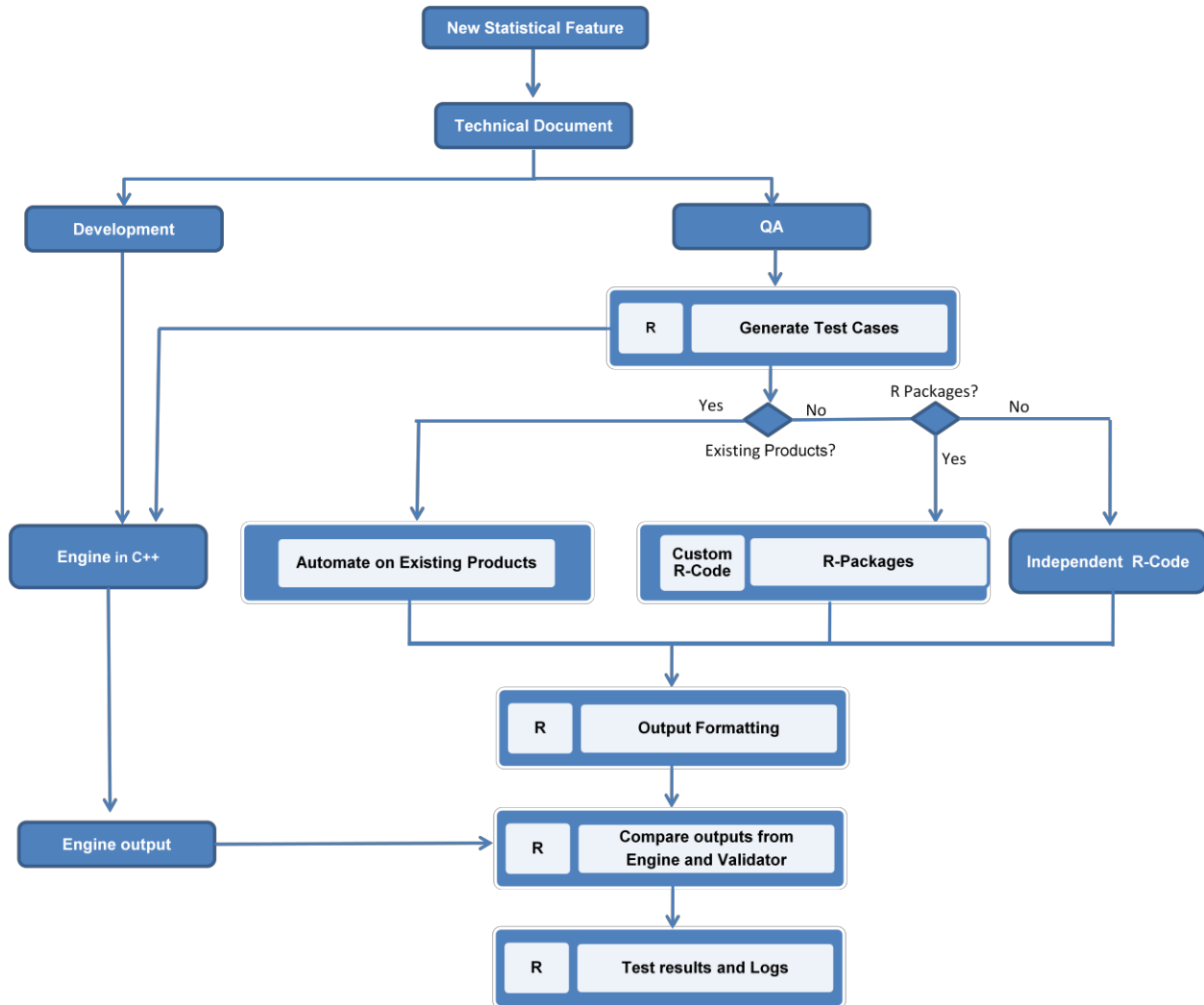


Figure 1: Statistical Algorithm Validation Flow

- A statistical feature requirement is considered for production. The subject matter expert studies the requirement and details it out in a technical specifications document. This document is then referred by developers and testers for development and validation respectively.
- Statisticians look for an external validator. This could be either own product or any other established validated software. There can be more than one product available for validation. The statistical validation chapter in the software user manuals describes the comparative validation performed on the statistical feature.
- If there are no external validators available, which is generally the case as the new development is the realization of innovative ideas, the underlying theory from research papers / articles is studied by statisticians.

PhUSE 2011

- A search through the repertoire of R packages is made for a package which has the same feature developed. It may be following a different algorithm or different implementation of the theory. If one or more R packages are of help, these are used in the validation.
- Occasionally, the set of R packages decided upon serves the purpose of validating the feature only partially. In such situations, R programs are written in-house following the same logic detailed out in the specification document.

R has extensive usage throughout all stages of the algorithm validation process. Specifically,

- **Generation of test cases:** Built in functions in R such as *rnorm()* which generate random numbers from different statistical distributions can be used for preparing test cases which serve as meaningful inputs to the algorithms. The combination of different parameter inputs help in coming up with exhaustive suite of test cases which ensure an effective coverage of all possible paths in the algorithm. Generation of extreme test cases can also be very well facilitated with the help of R.
- **Intermediate output:** User defined functions can be written in R to validate intermediate outputs. While testing a statistical algorithm, occasionally it is important to check the intermediate outputs traversing through the steps sequentially. This is very much possible with R being open source software. Even though one is using a readymade package in R, it is possible to insert statements within the code so as to get some extra or intermediate outputs. This is rarely possible with other packages.
- **Flexible Data Structure:** Extracting and storing the output in the desired format is convenient with R. A flexible data structure like 'List' is used very effectively for storing data in various forms.
- **Powerful Graphical representation:** Data visualization and graphics in R are of high quality. This is used in plotting the actual and expected output and comparing them. It helps in identifying patterns, trends, outliers etc which would not have been possible without graphics.
- **Automating the testing:** Since R interfaces well with Microsoft® Excel®, the test data stored in Excel is easily readable in R. Looping and case control in R make writing the script very easy. Automation, which is the key word in testing, is possible due to the Excel-R interface. Execution of the scripts is then independent of human intervention.
- **Comparing outputs:** The entire validation process can be automated if the outputs are generated as .CSV files and compared within R. This makes end to end automation of the testing possible.
- **Internal consistency checks:** In the absence of an R package, or inability to translate the same logic in R, which is generally the case if the algorithm is heavily dependent on simulations, one resort is to perform internal consistency checks with R. These checks are generally based on the intermediate outputs of the R program and the product under development. Obviously, one has to have an in-depth knowledge of the algorithm and has to come up with a proper testing strategy.
- **Improving the development of the algorithm:** If an algorithm is developed in languages like C++, while validating the same using R, it has been experienced that some of the parameters of the R program can be tweaked to see the effect on the outputs. This provides useful information on whether the scope of the algorithm can be extended to cover either better ranges of the parameters or can handle some additional options.
- **Checking the robustness of the algorithm:** Sensitivity analysis of the developed algorithm can be performed on the R validator but not on the developed engine as it is wrapped around by several validation checks. The independent R validator can be made totally free of the validation checks on the parameters thereby making the investigation on the robustness of statistical algorithm possible. Executing a variety of test cases with the R validator gives an indication of the robustness of the algorithm with respect to the parameters or methods used therein. This feedback is extremely important which can be shared with the users as validation reports as well as authentication of the procedures developed in Cytel products.
- **Calling R from products:** In majority of Cytel products, an interface to R is provided. The link to R can accept data from a product, call R routines written by user and display the output in HTML or the format used by the product. This facility enables users to compare the product outputs with the one given by their own developed R programs and convince themselves of the accuracy of the algorithms developed in the product.

STATISTICAL ANALYSIS SERVICES: VALIDATION OF CLINICAL REPORTING

SAFETY SUMMARIES

Safety summaries form the bulk of submissions in the data analysis services business at Cytel. A typical safety memo summarizes the parameters in the three safety datasets, namely, ECG, vital signs and laboratory tests. The routine memos involve computing the summary statistics of the parameters and plotting the corresponding graphs.

The analysis is carried out in SAS. A SAS macro is used to carry out the routine safety summaries. This macro produces the summary statistics of count, mean and the standard error for the parameter as well as for the change from baseline values.

PhUSE 2011

Validation of results is carried out by writing parallel and independent programs in R. An R script is generally used to develop such a program.

A function is written in R to validate the safety summaries. The R function inputs the raw SAS data, carries out the analysis and outputs the results in a .CSV file. The .CSV file is then read in SAS and compared with the summary datasets created in SAS. The same R function can be extended to input the SAS summary datasets and carry out the comparison within R itself.

Figure 2 illustrates the process:

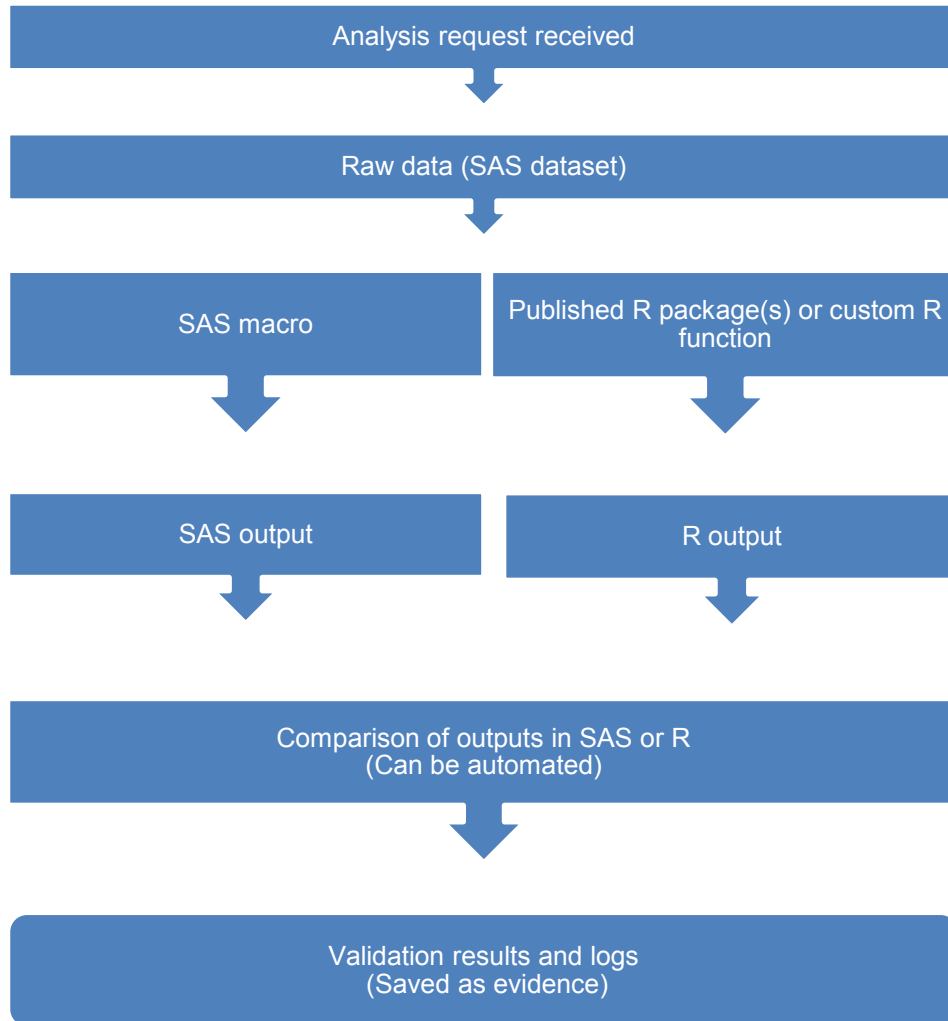


Figure 2: Clinical Reporting Validation Flow

The step-by-step process followed for a safety summary and how it is carried out in SAS and R is depicted in Figure 3.



Figure 3: Safety Summary Validation

PK SUMMARY, CLINICAL STUDY REPORTS AND AD-HOC REQUESTS

The pharmacokinetic (PK) summaries usually involve a mixed model. These are analyzed using the 'mixed' procedure in SAS. Traditionally, the validation was also carried out in SAS by a validator using an independent SAS code. However, R packages 'nlme' and 'lme4' are now being used to validate the results obtained from SAS. R software is also used to validate other summaries by writing independent code on a case by case basis for CSR and non standard or ad-hoc summaries.

THE CASE FOR THE 'OTHER' SOFTWARE

It is advisable to use another product to validate a statistical analysis carried out by the routinely used software for analysis and reporting. Such validation would be more reliable than that using the same package. A simple difference in the defaults employed across packages justifies the use of other software. As an instance, in SAS, a model with an intercept and a qualitative factor is defined in terms of the intercept and the indicator variables for all but the last level of the factor.³ However, in R the default is the "treatment" contrasts which are almost the same as the SAS parameterization except that they drop the indicator of the first level, not the last level. The validation program in different software needs to take these differences into account and thus provides more reliable validation.

AN ILLUSTRATION

A simple example can illustrate how the differences between the software help point out data peculiarities. An example of how different software can lead to mismatching output even with the same input data.

SAS code for the data generation:

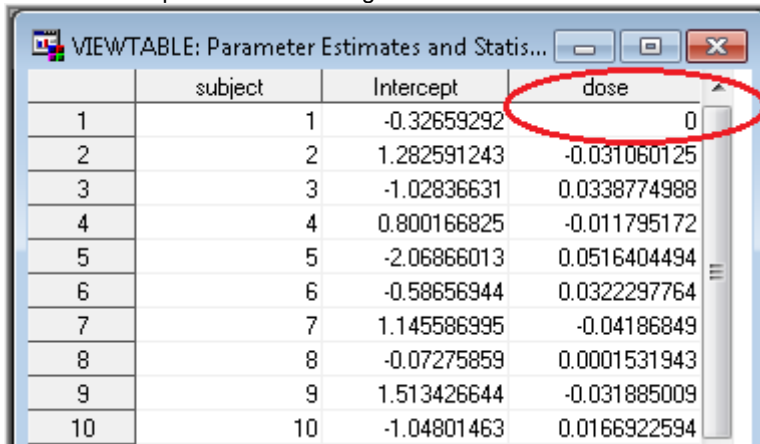
```
data phuse;
  seed = 123; /*Setting the seed to facilitate replication*/
  do subject = 1; /*Only one data-point for subject = 1*/
    do dose = 50;
      call rannor(seed, response);/*Random response for illustration*/
      output;
    end; /*dose*/
  end; /*subject*/

  do subject = 2 to 10; /*Data for other subjects with replicates*/
    do dose = 10 to 40 by 10;
      call rannor(seed, response);
      output;
    end; /*dose*/
  end; /*subject*/
run;

proc sort data = phuse;
  by subject;
run;

proc reg data = phuse outest = out noprint;
  model response = dose;
  by subject;
run;
```

Partial SAS output dataset showing the estimate of dose to be zero for subject =1:



	subject	Intercept	dose
1	1	-0.32659292	0
2	2	1.282591243	-0.031060125
3	3	-1.02836631	0.0338774988
4	4	0.800166825	-0.011795172
5	5	-2.06866013	0.0516404494
6	6	-0.58656944	0.0322297764
7	7	1.145586995	-0.04186849
8	8	-0.07275859	0.0001531943
9	9	1.513426644	-0.031885009
10	10	-1.04801463	0.0166922594

PhUSE 2011

Now, let us input the same data into R 2.13.1 and use the function 'lm' for linear model from the package 'nlme'

```
> library(nlme) # Invoking the package 'nlme'

> coeff_icept = as.vector (by (data = phuse, subject, function(x) lm (response ~ dose, data =
x)$coefficients [1])) # Saving the intercepts

> coeff_dose = as.vector (by (data = phuse, subject, function(x) lm (response ~ dose, data =
x)$coefficients [2])) # Saving the estimates for the dose

> coeff = data.frame (coeff_icept , coeff_dose)# Binding the vectors in a single 'data frame'
```

The output dataset:

	coeff_icept	coeff_dose
1	-0.32659	NA
2	1.282595	-0.0310602
3	-1.028365	0.0338774
4	0.800165	-0.0117951
5	-2.06866	0.0516405
6	-0.58657	0.0322297
7	1.14559	-0.0418685
8	-0.072755	0.0001532
9	1.513425	-0.031885
10	-1.048015	0.0166922

The outputs match except for the estimate of dose for subject = 1. (SAS gives a zero while R says NA) This little difference can lead to further differences in other statistics calculated using the estimates. It also provides an opportunity to inspect the data thoroughly and look for appropriate analytical methods.

STRENGTHS OF R

The R software has a number of important features and differences from SAS that make it a reliable validator. The availability of source code is the pre-eminent feature that allows customization and validation of the software to suit one's requirements. Apart from the open source nature, there are several advantages of using R software validating a statistical analysis or application.⁴

- R offers more analytical methods with the numerous packages implementing the latest in theory. The packages are validated by a set of validation tests maintained and upgraded by the R Core team⁵. This is very useful in validating first to market products. It takes a few years for new methods to be integrated in commercial software.
- Facility to run R from within other software allows access to latest in theory in a familiar environment while avoiding the cost of various add-on modules
- Updating your favorite packages in R is an easy task. A simple command 'update.packages' can be used to download and install packages which have an updated version available.
- It is very simple to add user defined functions in R. The ability to write your own completely integrated procedures in SAS or SPSS requires using a different language such as C or Python, and in the case of SAS, a developer's kit.
- R uses the objects NA and NaN (Not a Number) to store missing values and missing values due to impossible calculations respectively. Thus for a value of NA for the variable X, the comparison 'X < 0' returns a missing value unlike in SAS which returns a TRUE.
- R allows the use of variables from different 'data frames' or other data structures for analyses and graphics whereas in SAS, the data must be in a single dataset.
- R being an object oriented programming language enjoys all the benefits of OOPS.

ACCURACY OF R

One of the questions often raised is 'How accurate is R?' Like other software, R also has open forums in which the problems faced by R users are raised and tackled. One cannot say that R is completely free of errors. The comprehensive study by Keeling, et al⁶ (2007) compares nine statistics packages on the accuracy of modules such as univariate statistics, analysis of variance, linear regression, and non-linear regression. It was found that the accuracy of R was comparable to SAS and SPSS.

Bolker, Ben⁷ (2008) show that the accuracy of R has been improved. The very fact that R is constantly getting updated makes the latest development easily available and ensures that any accuracy issues are ironed out.

LIMITATIONS AND CHALLENGES

R is a great tool for the validation needed in the clinical trials domain. However, there could be some challenges in more wide spread use. For example,

- Help in R is more technical and not organized as compared that of other software.
- The numerous packages in R provide the latest in theory. However, the user needs to be aware of the packages and functions available.
- Frequent updation of the R packages might disturb the stability of the validator and make it difficult to maintain.
- The regulatory agencies and review officers need to be equipped with the necessary knowledge and the required software.
- There is a market perception to disregard free software as not being up to the mark.
- Performance wise, R is slower than other programming languages such as C++.
- R is inefficient in handling huge datasets and complex looping.

CONCLUSION

In conclusion, validation carried out using distinct software is highly advantageous and improves reliability. The R software is a useful tool to this end for validation of both, statistical software and analysis. The inherent capabilities and packages added by practicing statisticians and programmers make it an ideal choice for the validation of routine and advanced analysis as well as innovative, first to market products.

THE WAY AHEAD

Validating a graph is a tricky task. A primitive way could be to match the graphs visually to the corresponding tables. However, an exciting new method could be to use the R package 'digitize'⁸ to obtain the data behind the images created in SAS and other software, transform these into a dataset each and compare the datasets. This would make it all an objective process.

REFERENCES

1. <http://www.r-project.org/>
2. <http://www.fda.gov/downloads/RegulatoryInformation/Guidances/ucm126955.pdf>.
3. <http://cran.r-project.org/web/packages/SASmixed/vignettes/Usinglmer.pdf>
4. <http://www.springer.com/statistics/computational+statistics/book/978-1-4614-0684-6>
5. <http://www.r-project.org/doc/R-FDA.pdf>
6. Keeling, Kellie B. and Pavur, Robert J. A comparative study of the reliability of nine statistical software packages. 8, May 1, 2007, Computational Statistics & Data Analysis, Vol. 51, pp. 3811–3831.
7. Bolker, Ben. [R] software comparison. R Help Archive. [Online] 4 16, 2007. [Cited: 3 12, 2008.]: <http://finzi.psych.upenn.edu/R/Rhelp02a/archive/97802.html>.
8. <http://cran.r-project.org/web/packages/digitize/index.html>

ACKNOWLEDGEMENTS

We acknowledge Robert A. Muenchen for his book 'R for SAS and SPSS users' in the series 'Statistics in Computing by John Chambers David J. Hand, Wolfgang K. Hardle' which has been a very informative source of material for writing this paper. We also thank our colleagues in Quality Assurance and Services in Cytel, Pune for their suggestions and reviews of the paper.

CONTACT INFORMATION

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

Nikhil Abhyankar, Email: nikhil.abhyankar@cytel.com, Work Phone: +912066040146
Vidyagouri Prayag, Email: vidya.prayag@cytel.com, Work Phone: +912066040112

Cytel
8th floor, Siddhartha Towers, Kothrud,
Pune India 411029

Brand and product names are trademarks of their respective companies.