PharmaSUG China 2023 - Paper AA-116

Efficient Clinical Report Verification Workflow With QCfuns R Package

Yufan Chen, China R&D and Medical Affairs Janssen Research & Development

ABSTRACT

The pharmaceutical industry is transitioning from SAS to R and other open-source programming languages for clinical report programming. Many companies are encouraging the statistical programmers to practice R by validating the outputs produced by SAS. This paper presents a workflow for efficient clinical report validation using the self-developed R package QCfuns. A case study will be used to demonstrate the benefits of using the workflow and the package, including easier start, increased efficiency, and reduced risk of errors. Furthermore, the workflow is designed to enable new R programmers to accomplish validation tasks with ease, thereby promoting interest in the language.

INTRODUCTION

Clinical research generates a vast amount of data that needs to be collected, cleaned, and analyzed to produce accurate and reliable results. One critical aspect of clinical research is the creation of clinical reports that provide a summary of the study's objectives, methods, and results. The accuracy and integrity of clinical reports are essential for ensuring the safety and efficacy of investigational drugs.

Statistical programmers play a vital role in the production of clinical reports, and their expertise in data management, statistical analysis, and programming is critical to the success of clinical research. SAS has been widely accepted as the industry standard for generating clinical outputs in pharmaceutical industry, however, the popularity of R programming is increasing. Health authorities have also accepted the test submission package created by R, leading pharmaceutical companies to invest more in R. Although only very few studies are using R in the production work currently, there are a lot of other opportunities for statistical programmers to learn and practice R. To validate the accuracy of clinical reports, double programming is often required. Using R for validation provides statistical programmers with a chance to develop their skills in R programming through practice.

Efficient clinical report validation is a complex and time-consuming task that involves the identification of critical data elements, the execution of validation checks, and the resolution of validation issues. To facilitate this process, I have developed the QCfuns R package, which offers a range of functions for efficient clinical report validation.

In this paper, I will present a workflow for efficient clinical report validation using the QCfuns R package. I will demonstrate the application of the workflow using a case study and discuss the benefits of using the workflow and the QCfuns package in statistical programming. I believe that the workflow and the QCfuns package can help improve the accuracy and integrity of clinical reports and provide a valuable learning opportunity for statistical programmers who are transitioning to R programming.

METHODS

A typical workflow for clinical report validation comprises three main steps: (1) identification of critical data elements, (2) execution of validation checks and (3) resolution of validation issues.

- Step 1: Identification of critical data elements. The first step involves identifying the critical data elements that are essential to the clinical report's accuracy and integrity. These data elements may include demographic information, adverse events, laboratory results, and efficacy endpoints. When it comes to hands-on programming, you need to identify the correct ADaM/SDTM dataset and corresponding variables to use for analysis in this step.
- Step 2: Execution of validation checks. In step 2 of the validation process, the analysis dataset undergoes descriptive and statistical analysis to generate summary outputs. Following this, validation checks are executed by comparing the output created during this process with the one generated by the production side.

- Step 3: Resolution of validation issues. The final step involves resolving validation issues identified in the previous step.

Next, I will introduce some key features in the QCfuns package and explain how QCfuns enhances the validation workflow. The features include easy-to-start, simple function calls and validation results presentation.

EASY-TO-START

A code snippet in R is a small piece of code or a function that performs a specific task or calculation. It is typically written in the R programming language and can be executed within R environment such as RStudio. Code snippets can be shared and reused across projects and can be a valuable resource for statistical programmers to save time and increase efficiency.

The QCfuns package offers a convenient function called add_snippets, which allows you to effortlessly incorporate pre-built validation script templates for common validation tasks into your programming environment. These script templates provide an excellent starting point for validation work, as they contain comment sections and sample code that outline a typical validation workflow – from setting up the environment, to preprocessing validation data, generating outputs, and comparing them. Using code templates rather than starting from scratch simplifies the validation process and ensures that the script looks organized.

SIMPLE FUNCTION CALLS

The QCfuns package is a collection of R functions aiming to facilitate outputs validation process. The functions can interact with each other, and all together construct a workflow that is efficient and easy to follow.

There are two types of functions in the workflow as following.

- Data to table functions: These functions take a dataframe in ADaM structure as input and output a table-like dataframe. The names of these functions always start with "qc_" and end with the type of table they are creating, such as qc_demo, qc_shift.
- Utility functions: Utility functions are functions that perform tasks other than creating table-like dataframe from ADaM dataset. These functions include add_snippets, qc_rtf2df, qc_comparedf and qc_batchrun.

The QCfuns package employs a straightforward naming convention for its functions, using a combination of "qc_" and a brief description of their functionality. This approach offers a distinct advantage to users, as they can simply type "qc_" and quickly access all functions that begin with those letters. This intuitive naming convention eliminates the need for programmers to memorize specific function names, making it easier to navigate and utilize the package's full range of capabilities.

The QCfuns package’s data to table functions distinguishes itself from other table creation packages, such as tidytlg and rtable, by taking a unique approach. Rather than attempting to create highly adaptable functions capable of producing all types of tables, QCfuns employs a single, standardized table output function. For example, the qc_demo function streamlines the creation of a common demographic table by incorporating all necessary steps. By inputting the ADSL dataset and a variable list, a single call to qc_demo generates the desired table output. This approach utilizes wrapper functions to expedite the validation check in the second step of the typical validation workflow. Since each study requires a demographic table or summary of AE tables, a uniform table layout is shared across studies. However, this approach has its limitations, and customized methods are required for study-specific tables. To meet these specific requirements, the QCfuns package also includes more versatile functions like qc_cat_row and qc_num_row.

VALIDATION RESULTS PRESENTATION

The QCfuns package offers several functions that facilitate the generation of summary reports of the validation results, streamlining the identification and resolution of validation issues. For instance, the

qc_comparedf function leverages the compare_df function from the compareDF package to generate a concise and color-coded summary of comparison results between the data frame created and the data frame extracted from the rtf file (Display 1). Similarly, the qc_batchrun function can digest the results from qc_comparedf and produce a comprehensive summary of the validation status for multiple outputs simultaneously (Display 2) with log and comparison output for each individual script. The qc_batchrun can be executable not only in RStudio interactively, but also allow users to run batch jobs independently of the RStudio IDE.

Comparison Results for test_output2

row_seq	chng_type	Row_text	Treatment	Placebo
1	PROD	N	6	10
1	QC	N	5	10

Display 1. Screenshot of qc_comparedf Result

Comparison Results

Output ID	All Matched	Error	Warning
tefwho02a	No	object 'tab1' not found	
tefwho01	No		
tefwho02	No		
tsidem02	Yes		
tsidem02s1	Yes		
tsidem02s2	Yes		

Display 2. Screenshot of qc_batchrun Result

In summary, the workflow for efficient clinical report validation using the QCfuns R package provides a systematic approach to clinical report validation and enables efficient identification and resolution of validation issues. The QCfuns package provides a set of functions that can help improve the accuracy and integrity of clinical reports and facilitate the transition to R for clinical report programming.

RESULTS

The QCfuns R package workflow has been successfully integrated into the company's programming practices, eliciting positive feedback from its users due to its speed and user-friendliness. To showcase the efficacy of this workflow and package, a hypothetical case study to validate a clinical report for a mock clinical trial will be demonstrated during the presentation, which comprises of demographic data, adverse events, and laboratory results.

CONCLUSION

The QCfuns R package provides a fast and efficient way to execute validation checks and generate a summary report, which helps us quickly identify and resolve the validation issues. The package also provides a set of flexible functions that can be easily customized to specific validation requirements, which can improve the accuracy and integrity of clinical reports.

One limitation of the package is that it is based on J&J conventions, and the read in rtf file process may not be generalizable to other clinical reports created by other companies. Additionally, while the QCfuns package is designed to streamline the validation process, it still requires a significant amount of

programming expertise to use effectively. Therefore, additional training and support may be needed to fully implement the package in a clinical report validation programming workflow.

In future work, we plan to evaluate the performance of the workflow and the QCfuns package in a larger sample of clinical reports, and to explore ways to further automate and streamline the validation process. We also plan to expand the functionality of the QCfuns package to include additional validation checks and reporting capabilities.

Overall, the workflow and the QCfuns R package offer a promising approach to clinical report validation that can improve the accuracy and integrity of clinical reports while reducing the time and effort required for validation. As the pharmaceutical industry shifts from SAS to R and other open-source programming languages, the QCfuns package provides a valuable tool for statistical programmers to enhance their R programming skills and improve their clinical report programming ability.

REFERENCES

Roche PD China Data Sciences. Good Software Engineering Practice for R Packages. March 2023. https://openpharma.github.io/workshop-r-swe/

Pang, Carlos. Practices for Comparing R Data Frames with SAS Outputs. China Pharma RUG Meeting March 2023.

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Mori, Kota. striprtf: Extract Text from RTF File. https://CRAN.R-project.org/package=striprtf

Joseph, Alex. compareDF: Do a Git Style Diff of the Rows Between Two Dataframes with Similar Structure. https://CRAN.R-project.org/package=compareDF

ACKNOWLEDGMENTS

Thanks for the support and valuable feedback from David Fan, Casey Yang, Yingshan You, Sheng-Wei Wang and many other colleagues from statistical programming team of Janssen R&D.

RECOMMENDED READING

QCfuns Github Repository: https://github.com/YufanC/QCfuns

CONTACT INFORMATION

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

Yufan Chen
China R&D and Medical Affairs Janssen Research & Development
ychen209@its.jnj.com