Paper ETD02

Efficient TFL Review within Statistical Programming – Using R and SAS

Siddharth Kumar, Navitas Data Sciences, Pottstown, USA

ABSTRACT

In clinical data analysis, while all outputs undergo a process of independent QC involving double programming, certain issues in RTF (Rich Text Format) outputs may not be detected by PROC COMPARE, resulting in unaddressed quality issues that are later identified during the cross-functional review. To instill confidence in the TLF (Tables, Listings, and Figures) package, a thorough review within the statistical programming team is essential. Manually reviewing over ~300 RTF outputs in a submission package is time-consuming and prone to errors. This presentation addresses these challenges by proposing an automated solution using R and various R packages to streamline the review process.
The traditional manual review involves verifying numerous aspects such as population counts (Big N), counts for primary and secondary endpoints, adverse events, cross-checking outputs, decimal precision, alignment, spell checks, titles, footnotes, page breaks, ensuring validation program is run after production program, etc. Relying on manual checks across many files introduces inefficiencies and the potential for human error. To overcome these challenges, automated workflow in R and SAS which systematically processes and analyzes RTF files and SAS datasets is helpful.

INTRODUCTION

In clinical data analysis, generating Tables, Listings, and Figures (TLF) for Clinical Study Reports (CSR) is a crucial step in preparing regulatory submissions. The validation of these outputs serves as a guarantor process, ensuring the accuracy and reliability of trial results, and maintaining the reputation of sponsors and Contract Research Organizations (CROs). Double programming has long been the gold standard for validation, providing a robust mechanism for cross-checking results and mitigating errors. These activities are often outsourced to CROs, requiring sponsors to perform a tertiary QC to validate and ensure the quality of the outputs. Since timelines are often compressed, statistical programming teams must find ways to enhance both efficiency and accuracy.
A similar challenge exists within statistical programming teams, balancing multiple deliverables while maintaining quality can be challenging. Before sharing TLF outputs with cross-functional teams, an internal quality review is essential, particularly for output (RTF) files. While PROC COMPARE is effective in verifying data, it fails to detect issues within RTF files that arise at the PROC REPORT stage, such as counts appearing in incorrect columns, switched population (Big N) values, and formatting or content-related issues like misaligned text, page break inconsistencies, and missing footnotes. Additionally, it is crucial to cross-check outputs for consistency in decimals, alignment, and overall formatting. These undetected errors can lead to quality concerns from reviewers, potentially delaying submissions and requiring rework.
Therefore, a manual review of all RTF files remains essential to ensure accuracy and quality. This review is often performed by the lead programmer, and it is a tedious and time-consuming process. Currently, manual review remains the standard approach for checking RTF outputs. This involves verifying population counts (Big N), counts for primary and secondary endpoints, adverse events, decimal precision, text alignment, spell checks, cross-referencing outputs, reviewing titles, footnotes, etc. However, with hundreds of outputs to review, this process is time-consuming, tedious, and prone to human error.
To streamline and enhance this process, an automated workflow using R and SAS can serve as a transformative solution. By systematically analyzing both RTF files and SAS datasets, automation can take over many of the labor-intensive checks, reducing manual effort while improving accuracy and consistency. Leveraging R's specialized packages and programming techniques, this solution ensures a more efficient and error-free review process, ultimately enhancing the quality of TLF packages for regulatory submissions.

By automating critical iterative tasks, statistical programmers can free up valuable time to focus on more complex aspects of clinical trials, such as primary efficacy endpoints, advanced algorithms, and in-depth data analyses. This shift not only improves efficiency but also strengthens the reliability and reproducibility of clinical trial results, ensuring that the truth of the data is accurately conveyed—paving the way for successful regulatory submissions.

Traditional QC

In clinical data analysis, traditionally, all outputs undergo independent QC through double programming, which involves recreating the final SAS dataset. Typically, the production programmer generates the final dataset, while the QC programmer independently replicates it using the same variable names and structure before performing a PROC COMPARE to ensure consistency. However, this approach focuses solely on data validation and does not account for potential presentation errors in RTF outputs which can occur at PROC REPORT stage. As a result, a manual QC of RTF outputs is required to identify issues such as values appearing in incorrect columns, misalignment, page break inconsistencies, and missing footnotes. This manual review process is time-consuming and prone to human error, highlighting the need for a more robust approach.

During the manual review, the lead programmer must perform a series of critical checks to ensure the accuracy and quality of the final outputs. There are a number of checks that need to be performed during the manual review. Here are the commonly used 10 check points.

Proposed Solution

This paper proposes an efficient solution using R and SAS to streamline the RTF review process, ensuring accuracy and consistency across the Tables, Listings, and Figures (TFL) package.

Step 1:

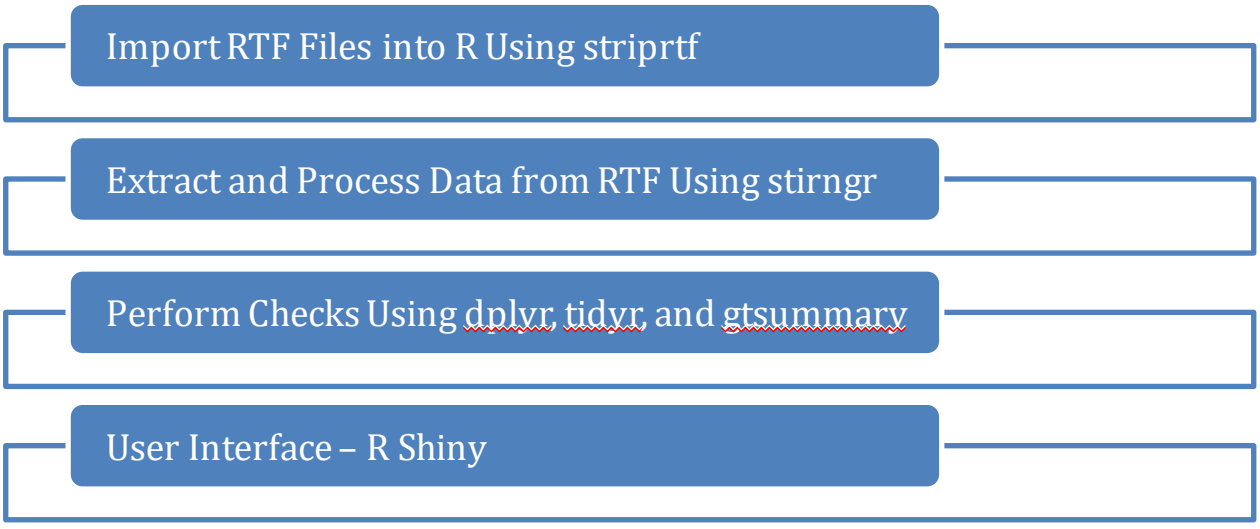
To enhance QC efficiency, convert RTF outputs into SAS datasets, allowing QC programmers to validate the final output rather than just the underlying dataset. This approach ensures that discrepancies between the final dataset produced before PROC REPORT stage and the RTF output are addressed. Import RTF files to SAS and programmatic QC of dataset reduces risk and improves quality. It eliminates need for variable naming convention.

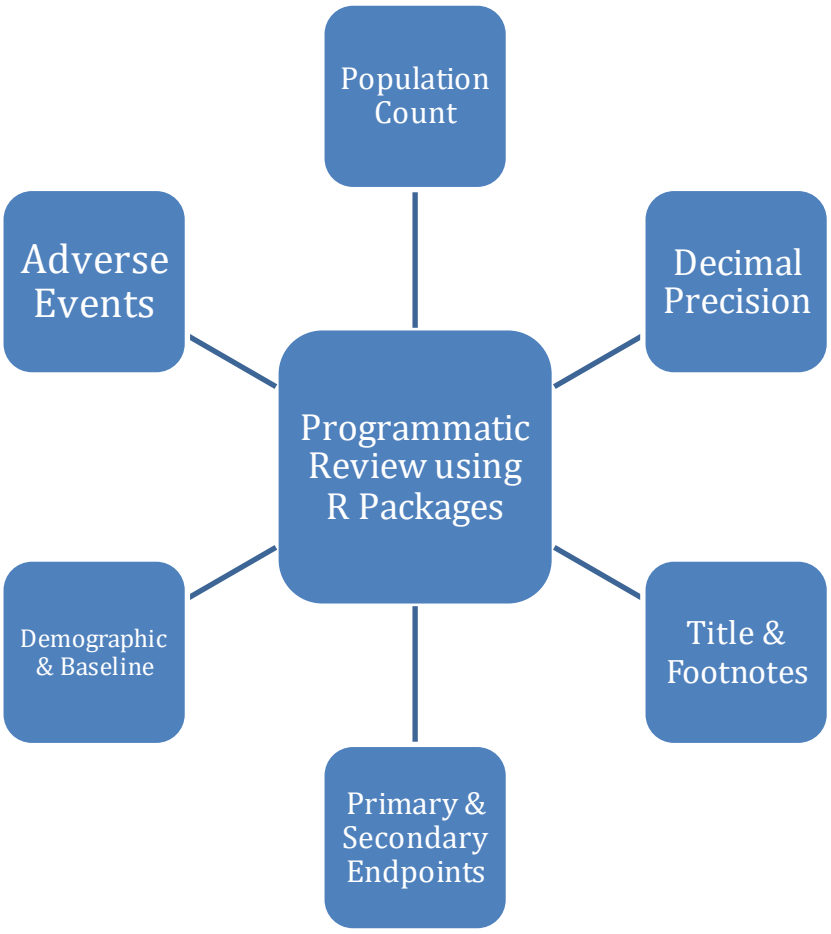
Step 2:

Replace manual review with R programming checks. Read RTF files into R using STRIPRTF, UNRTF packages and manipulate using dplyr, Tidyverse, and output using shiny. In the below example, I am demonstrating the checks for Big N, and decimal precision. Using the same approach, checks can be created for all the manual review tasks and then a combined output generated either in Excel or Smartsheet would help facilitate the tables and listings package review.
Additionally, R provides several packages for end-to-end clinical trial analysis, including tools for generating tables and listings. Implementing automated workflows that incorporate both data and output validation can significantly improve efficiency and accuracy in clinical data reporting.
To achieve this, R programming is utilized, leveraging several powerful packages for data manipulation and RTF file handling. Below is a list of the R libraries used in this process:

R Packages	Description
Shiny	Build interactive web applications with R.
striprtf	Extract Text from RTF File
stringr	Consistent wrappers for common string operations
dplyr	Tool for working with data frame like objects
writexl	Export data frames to excel 'xlsx' format
gtsummary	Creates table summarizing datasets, regression models and more.
hunspell	Spell checker

Workflow





Scenario 1 – Population Count

Population counts are assigned during the PROC REPORT stage in SAS. There is a possibility of incorrect presentation of these counts—such as treatment population counts (Big N) being assigned to the wrong treatment group (e.g., Placebo Big N showing up in the treatment group and vice versa). This issue, though typically found in a few outputs within a large package of approximately 300 tables, requires a tedious manual review of all outputs. This task can be automated using R programming by extracting the Big N values from RTF outputs to ensure that the population counts are presented correctly for each treatment group. The process is streamlined and automated using powerful R packages, including striprtf, stringr, dplyr, tidyr, and Shiny. The user interface is created using Shiny, allowing users to input the folder path containing RTF files, input the treatment names and initiate the validation process. The results are displayed within the Shiny app and can be downloaded as an Excel file for further review. This automation significantly reduces the time and effort required for manual validation, improving efficiency and accuracy.

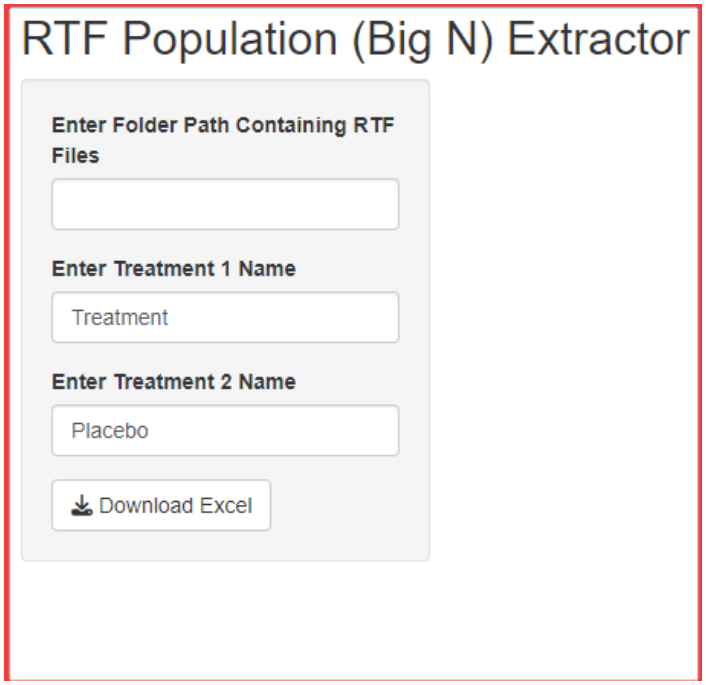
R Program –

```
library(stringr)
library(striprtf)
library(dplyr)

rtf_text <- read_rtf ("C:/Users/kumars/Documents/TLF/AP/t_14_1_1_1_disp.rtf")
rtf_text ## Character Vector

extracted_T1 <- str_extract(rtf_text, "Treatment[^)]*\\)")
extracted_T1

extracted_T2 <- str_extract(rtf_text, "Placebo[^)]*\\)")
extracted_T2
```

```
 [1] "*|  | "
 [2] "*| Visit |  | Treatment\n(N=71) | Placebo\n(N=101) | "
 [3] "*|  | "
 [4] "*| Visit 2 |  |  |  | "
 [5] "*|   Very Much Better |  |     6 ( 10%) |     2 (  4%) | "
 [6] "*|   Moderately Better |  |    13 ( 22%) |     3 (  6%) | "
 [7] "*|   A Little Better |  |    14 ( 24%) |     4 (  8%) | "
 [8] "*|   No Chagne |  |    10 ( 17%) |    28 ( 55%) | "
 [9] "*|   A Little Worse |  |    10 ( 17%) |    12 ( 24%) | "
[10] "*|   Moderately Worse |  |     3 (  5%) |     1 (  2%) | "
[11] "*|   Very Much Worse |  |     2 (  3%) |     1 (  2%) | "
[12] "*|  | "
[13] "*| Visit 3 |  |  |  | "
[14] "*|   Very Much Better |  |    10 ( 19%) |     1 (  2%) | "
[15] "*|   Moderately Better |  |    13 ( 25%) |     5 (  8%) | "
[16] "*|   A Little Better |  |     8 ( 15%) |    12 ( 20%) | "
[17] "*|   No Chagne |  |    10 ( 19%) |    28 ( 47%) | "
[18] "*|   A Little Worse |  |     9 ( 17%) |     9 ( 15%) | "
```

R Shiny – User Interface is developed using R shiny to read all the RTF files from a specified folder, processes the content, and displays results within the app. Users can review the output interactively and download the results as an Excel file for further analysis.

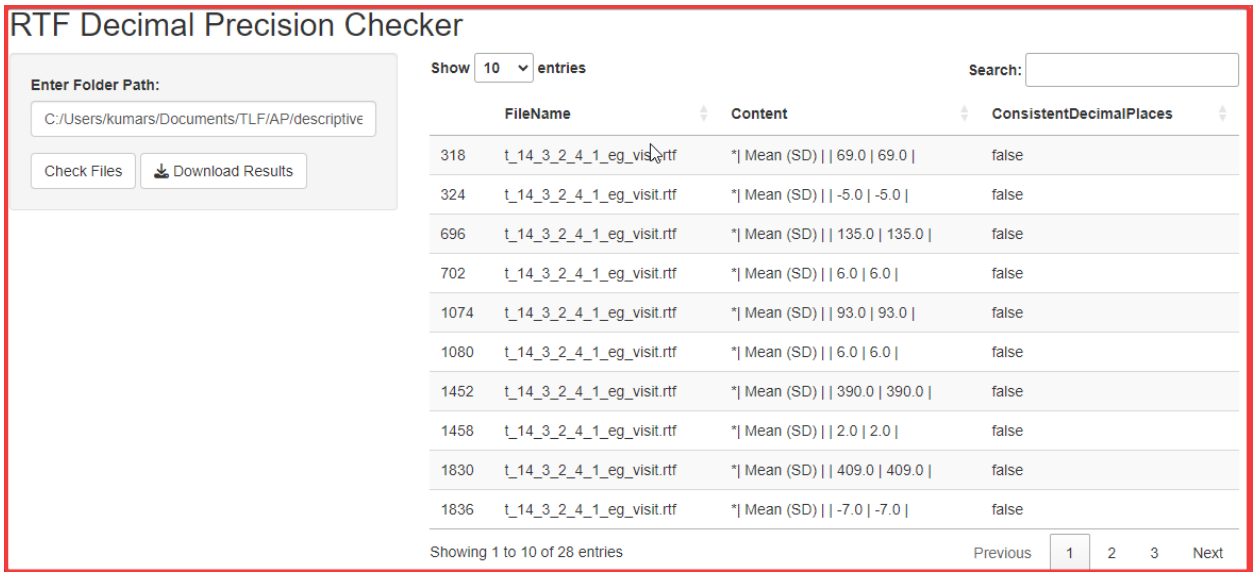
```
ui <- fluidPage(
  titlePanel("RTF Big N Extractor"),
  sidebarLayout(
    sidebarPanel(
      fileInput("folder", "Select Folder Containing RTF Files", multiple = TRUE, accept = ".rtf"),
      textInput("treatment1", "Enter Treatment 1 Name", value = "Treatment"),
      textInput("treatment2", "Enter Treatment 2 Name", value = "Placebo"),
      downloadButton("downloadData", "Download Excel")
    ),
    mainPanel(
      tableOutput("extractedData")
    )
  )
)
```



...3	...6	...11	...12	C	rtfnan	nam	Treatment	able	Placebo
Table	t_14_1_2_	Table 14.1.2.	Demographics and Other Baseline Char	C2	F:/Biome	c2	Treatment (N= 71)	c2	Placebo (N=100)
Table	t_14_2_1_	Table 14.2.1.	Progression Free Survival by Subgroups	C20	F:/Biome	c20	Treatment (N= 71)	c20	Placebo (N=100)
Table	t_14_2_2_	Table 14.2.2.	Objective Response Rate by Subgroups	C22	F:/Biome	c22	Treatment (N= 71)	c22	Placebo (N=100)
Table	t_14_2_13_	Table 14.2.13	Summary of Participant Global Impress	C57	F:/Biome	c57	Treatment (N= 71)	c57	Placebo (N=100)
Table	t_14_2_14_	Table 14.2.14	Summary of Participant Global Impress	C58	F:/Biome	c58	Treatment (N= 71)	c58	Placebo (N=100)
Table	t_14_3_1_	Table 14.3.1.	Overall Summary of Treatment-Emerge	C59	F:/Biome	c59	Treatment (N= 71)	c59	Placebo (N=100)
Table	t_14_3_1_	Table 14.3.1.	Treatment-Emergent Adverse Events b	C60	F:/Biome	c60	Treatment (N= 71)	c60	Placebo (N=100)
Table	t_14_3_1_	Table 14.3.1.	Treatment-Emergent Adverse Events R	C61	F:/Biome	c61	Treatment (N= 71)	c61	Placebo (N=100)
Table	t_14_3_1_	Table 14.3.1.	Serious Treatment-Emergent Adverse E	C62	F:/Biome	c62	Treatment (N= 71)	c62	Placebo (N=100)

Scenario 2

Maintaining decimal consistency as defined in the Statistical Analysis Plan (SAP) is crucial. The lead programmer's task is to verify that all outputs adhere to the decimal precision specified in the SAP. This manual review can be replaced with a programmatic check using the approach outlined above. By leveraging the R package Striprtf to read RTF files into R, followed by manipulation using stringr, dplyr, and shiny, the program can efficiently check for decimal precision consistency and generate the necessary output, streamlining the review process and ensuring consistency. In this case, the prespecified decimal precision from the SAP is incorporated into the R code. This allows the program to compare the actual decimal precision in the outputs and flag any discrepancies. By automating this process, the lead programmer can efficiently verify decimal consistency across numerous outputs, minimizing manual checks and providing quicker results.



Scenario 3

It is essential to ensure that titles and footnotes in RTF outputs are consistent, accurate, and error-free. Discrepancies in titles, spelling mistakes, or unresolved macros can significantly impact the quality of the final output. The proposed R programming solution automates this manual task, utilizing several packages, including hunspell, striprtf, stringr, dplyr, and shiny.

By incorporating checks, the program can flag any unresolved macros within the RTF files. It also performs a spell check using the hunspell package to detect and correct spelling errors. Additionally, the program provides a user interface through shiny, allowing the user to input the folder path where all RTF outputs are saved. The program processes these files and generates a report that can be downloaded as an Excel file for easy review. This automated solution significantly reduces manual review time, minimizes the risk of errors, and ensures that the final RTF outputs meet the required quality standards.

File	Content	SpellCheck
t_14_3_1_1_ae_overall.rtf	AE: Adverse Event; AESI: Adverse Event of Special Interest; CTCAE: Common Terminology Criteria for Adverse Events; MedDRA: Medical Dictionary of Regulatory Activities; NCI: National Cancer Institute;	AE, AESI, CTCAE, MedDRA, NCI
t_14_3_1_1_ae_overall.rtf	TEAE: Treatment-Emergent Adverse Event; SAE: Serious Adverse Event.	TEAE, SAE
t_14_3_1_1_ae_overall.rtf	N represents the number of participants in the corresponding cohort.	No errors
t_14_3_1_1_ae_overall.rtf	All AEs were coded using MedDRA Version 24.0.	AEs, MedDRA
t_14_3_1_1_ae_overall.rtf	The severity of All AEs are graded using NCI-CTCAE Version 5.0: Grade 1=Mild; Grade 2=Moderate; Grade 3=Severe; Grade 4=Life-threatening; Grade 5=Death.	AEs, NCI, CTCAE
t_14_3_1_1_ae_overall.rtf	TEAEs are classified as study drug-related if they have an investigator determination of related to study treatment.	TEAEs
t_14_3_1_1_ae_overall.rtf	A TEAE is defined as an AE that starts or worsens on or after the first dose of study treatment is taken, including those occurring or increasing in severity up to 30 days after the last dose of study treatment.	TEAE, AE
t_14_3_1_1_ae_overall.rtf	A TEAE is defined as leading to a dose modification if the action taken with study treatment for a TEAE was having a dose interruption, dose reduction or dose increase.	TEAE, TEAE
t_14_3_1_1_ae_overall.rtf		No errors

ENHANCEMENTS

Using the same approach, other manual checks for adverse event counts, key demographics, baseline characteristics, primary and secondary endpoints can be automated. This reduces time while ensuring

accuracy and consistency. In future, a Generative AI-powered app could further enhance efficiency, provided that data security and access controls are carefully implemented.

CONCLUSION

Efficient review and validation of Tables, Listings, and Figures (TFLs) are critical in the clinical trial analysis process to ensure that results are accurate, consistent, and comply with predefined standards. Traditional manual validation methods, while effective, are time-consuming and prone to human error, especially when dealing with large numbers of outputs. This paper demonstrates how automating the review process using R and SAS can significantly enhance the efficiency and accuracy of TFL validation. By leveraging R packages like striprtf, dplyr, stringr, and shiny, it is possible to automate the extraction of key data points from RTF files, ensuring enhanced output quality. The integration of these tools into a streamlined workflow allows for faster, more reliable validation, minimizing manual effort and reducing the risk of errors. The use of a Shiny app for user interaction simplifies the process, enabling easy data input, visual feedback, and downloadable results for comprehensive review. Overall, the automation of the TFL review process not only saves time but also improves the quality and reliability of the output. The approach presented in this paper offers a scalable solution for handling large-scale clinical trial data and can be easily adapted to meet the specific needs of different organizations and projects. By combining the strengths of both R and SAS, statistical programming teams can achieve a higher level of efficiency and precision in their TFL review processes.

REFERENCES

R: A language and environment for Statistical Computing

Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2024). _shiny: Web Application Framework for R_. R package version 1.10.0, https://CRAN.R-project.org/package=shiny

Mori K (2023). _striprtf: Extract Text from RTF File_. R package version 0.6.0, https://CRAN.R-project.org/package=striprtf

CONTACT INFORMATION

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

Author Name: Siddharth Kumar Mogra
Company: Navitas Data Sciences
Address: 1610 Medical Drive #300, Pottstown, PA 19464
Work Phone: 1-484-366-7644
Email: siddharth.kumar@navitaslifesciences.com
Website: www.navitaslifesciences.com