Paper SD08

Data extraction and Graphical visualization for quick decision making

Mukul Mittal, Sanofi, Bridgewater, NJ, USA

Andre Couturier, Sanofi, Bridgewater, NJ, USA

ABSTRACT:

Summary tables from clinical trials offer a succinct overview of key findings, including participants demographics, treatment effects, and safety outcomes. While these tables provide essential insights, visualization of the data significantly enhances comprehension and communication. However, publications and older studies often present summarized data without raw data access, complicating accurate trend analysis. To address this, we developed R-Shiny applications, **Data ExtractoR** to extract data from summary reports in formats such as PDF/RTF/DOCX and **Graph DesigneR** for visualization. Our solution incorporates a Python environment with the **Camelot** package for PDF table extraction, **docxtractr** for word, and **striprtf** package for RTF files. Users can select and process multiple tables, and the data undergoes automated cleaning operations through custom functions. The cleaned data can then be visualized using **Graph DesigneR app**, based on **ggplot2** package, enabling quick insights and trend analysis. This approach streamlines the process, reduces errors, and enhances efficiency.

INTRODUCTION:

Data extraction is the process of retrieving relevant data from published studies or summary reports in an automated way to gain more insights, make faster decisions and support 'Data Story Telling'. In clinical trials data is generally presented in a summary report e.g., absolute count and percentage or min, max, q1, q3 etc. Presentation of the data in this form help reviewers to understand the study trend but takes lots of time to interpret. On the other hand, data published in journals or conference proceedings give you a glimpse of overall trend, but these reports or data cannot be used for other purposes. Graphical visualization of data and information like charts, graphs and maps etc. provide an easy way to see and understand trends, outliers and patterns in no time and data can be shown in a concise and compact way.

To address these issues and challenges, Analysis and Reporting Technologies (ART) team at Sanofi developed two applications: Data ExtractoR and Graph DesigneR to retrieve data from published reports in PDF/RTF/DOCX format and "massage" data e.g., variable renaming, wide to long or long to wide form transformation etc. Once extracted, the data is transformed in desired way and fed into the Graph DesigneR to create graphs from scratch or using our template library. This process of data extraction and graphic visualization using these applications saves time and is less prone to errors by avoiding copy/paste and manual data entry. Instead, the tool automates and simplify data extraction with a no-code interface to prepare publication data for visualization. This will immensely help users to visualize summarized data in graphical format.

METHODOLOGY AND RESULTS:

IMPORTING SUMMARY REPORT FILES:

Data extraction begins with the crucial step of importing tables and/or reports into the application. **Data ExtractoR** is designed to support a variety of file formats e.g., PDF/RTF/DOCX, ensuring flexibility and efficiency (Fig. 1).

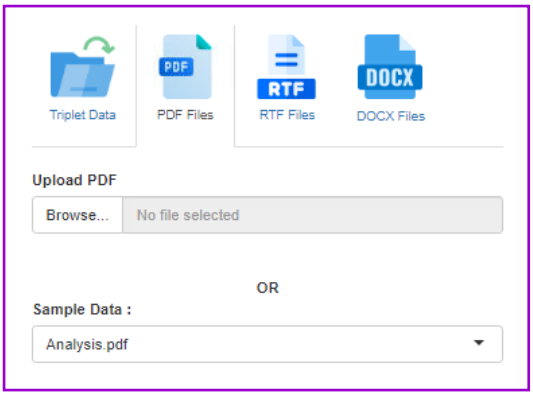

Fig.1: Upload Data - Application UI

TABLE/PAGE SELECTION

The Table/Page Selection tab streamlines the process of organizing and managing extracted tables by separating them sequentially (e.g., Page 1, Page 2) regardless of the source. For DOCX files, tables are extracted using functions from the **officer** or ***docxtractr*** packages, which identify table structures and return them as data frames. For RTF files, the ***striprtf*** package splits tables based on predefined patterns, ensuring each table is parsed individually. For PDF files, ***Camelot*** leverages its **read_pdf** function to detect and extract tables, separating them into a list by identifying table boundaries on each page (Fig. 2).

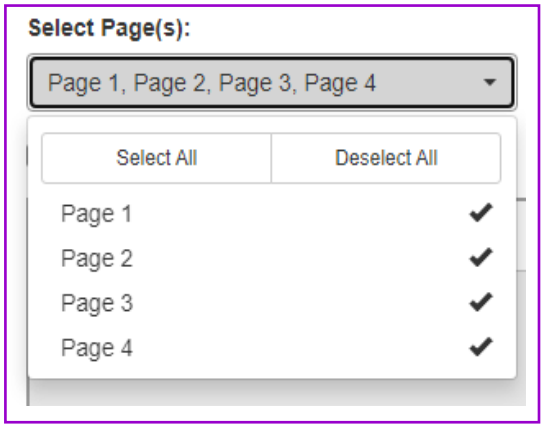

Fig.2: Table/Page Selection

HEADER ROW(S) DETECTION

Clinical tables generally have single and multiline headers. To manage multi header tables we created a custom function **"Merge Header Automatically"** to merge fragmented header information together and allow users to rename the resulting columns into a single, cohesive structure in Data ExtractoR. This process utilizes the ***stringr*** package, specifically functions like **str_detect,** to identify patterns and filter out irrelevant rows. Using the ***dplyr*** package, functions such as **mutate** and **filter** are applied to transform and refine header rows efficiently. Figures 3.1 shows a multi-header table and Figure 3.2 shows the result when the Merge Header Automatically is triggered.

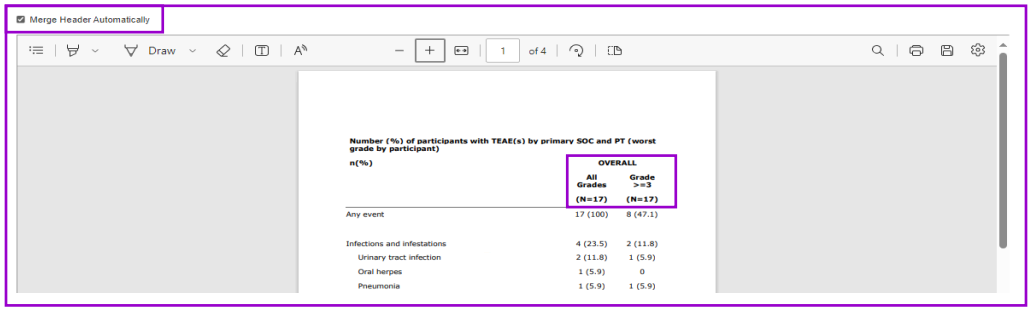

Fig.3.1: Merge Header Automatically Checkbox with Preview

	V1	V2	V3
1	Select/Unselect All		
2	n(%)	OVERALL_All_Grades_(N=17)	OVERALL_Grade_>=3_(N=17)
3	Any event	17 (100)	8 (47.1)
4	Infections and infestations	4 (23.5)	2 (11.8)
5	Urinary tract infection	2 (11.8)	1 (5.9)
6	Oral herpes	1 (5.9)	0
7	Pneumonia	1 (5.9)	1 (5.9)

Fig.3.2: Merged Header

DATA CLEANING, COLUMN REMOVAL, AND RENAMING:

The Data Cleaning Tab is designed to streamline dataset refinement through both custom and automatic modes. In custom cleaning mode, users can select specific transformations from a predefined list of functions and apply them interactively to address unique dataset needs with great flexibility. Available transformations include removing blank/invalid rows, split rows with parentheses into the next row, parsing confidence intervals values, split semicolon/colon/comma values enclosed in parentheses, moving bracketed values, adding new columns for bracketed values, create numeric columns for bracketed values and removing parentheses.

The automatic cleaning mode further streamlines the process by intuitively detecting and applying many of these same transformations without user intervention, based on the dataset's structure. For instance, it identifies and removes blank rows, splits the first column if delimiters are present, parse confidence intervals values, relocates bracketed values, and adds new bracket-based columns automatically, leveraging the same underlying functions (Fig. 4).

Each transformation leverages numerous R packages, including **dplyr** and **stringr**, to manipulate and clean data effectively. Functions such as **mutate, filter,** and **str_detect** are central to these operations, ensuring that datasets are transformed dynamically and accurately.

Fig.4: Data Cleaning

DATA RESHAPING AND MANIPULATION:

Data reshaping is a critical step in the processing of clinical trial data, where programmers need to reshape data (transforming from **wide to long** or vice versa), and derive new variables based on multiple **if else** conditions. This tasks also encompass altering the **case of text or patterns**, which is done by leveraging R packages like ***stringr*** for string manipulation, ***dplyr*** for adding new variables, ***tidyr*** for data reshaping. We implemented these functionalities in Data Manipulation tab of **Data ExtractoR** (Fig. 5).

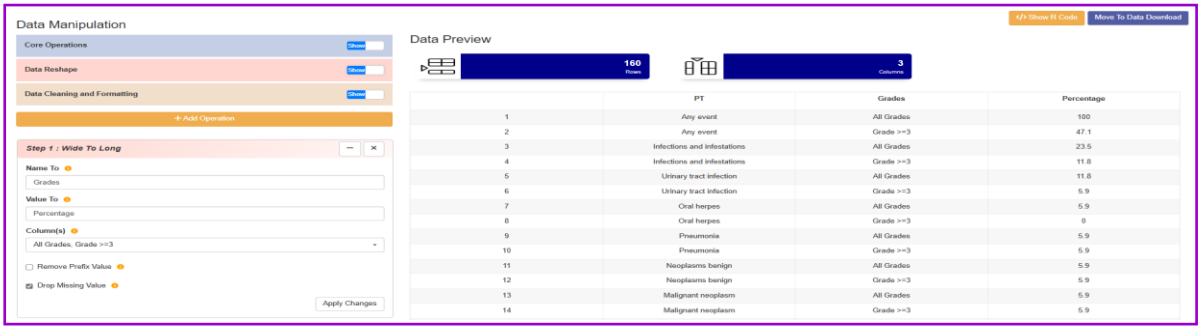

Fig.5: Data transposition- wide to long reshaping

DATA DOWNLOADING: The Download Data tab supports exporting transformed datasets in multiple format such as CSV, Excel, XPT, JSON, and SAS-compatible files, using different packages (Fig. 6). For SAS (.sas7bdat files), the application leverages the ***SASPy*** Python library, through an integrated workflow that converts R data frames into pandas data frames using the reticulate package producing datasets where column metadata and formatting is preserved. This ensures compliance with SAS standards, maintaining data integrity and compatibility with industry-standard analytical tools widely used in clinical trials and regulatory reporting. Once downloaded, users can upload these files to the Graph DesigneR or other visualization tools for in-depth exploration of trends and patterns in their data or build powerful Data Story Telling visualizations.

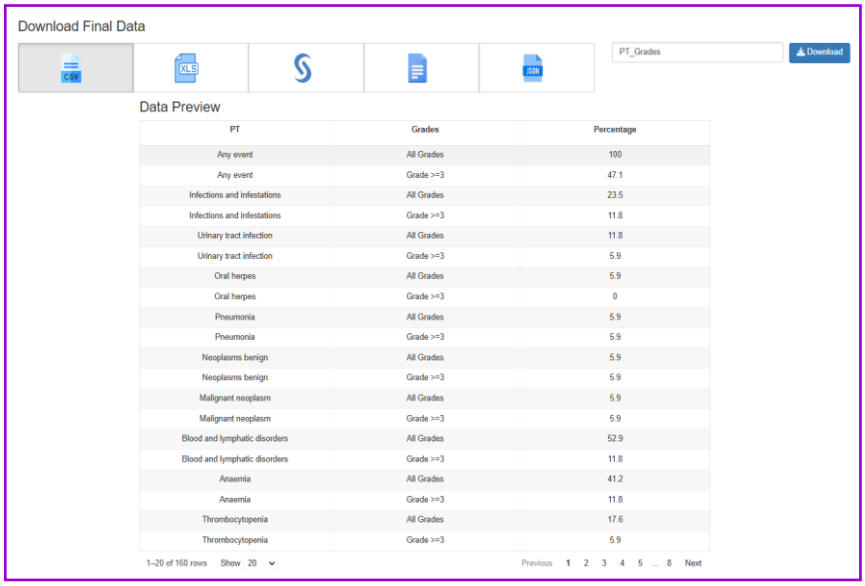

Fig.6: Data Download

CODE GENERATION:

The Download Data tab not only allows users to export datasets in various formats, but it also dynamically generates an R script reflecting all transformations performed across the application. These transformations i.e., data cleaning, header manipulation, copy/paste operations are captured in code snippets built using ***rlang*** package (Fig. 7). The ***rlang*** approach lets the app create unevaluated expressions on the fly, ensuring that every user action is accurately translated into an R command.

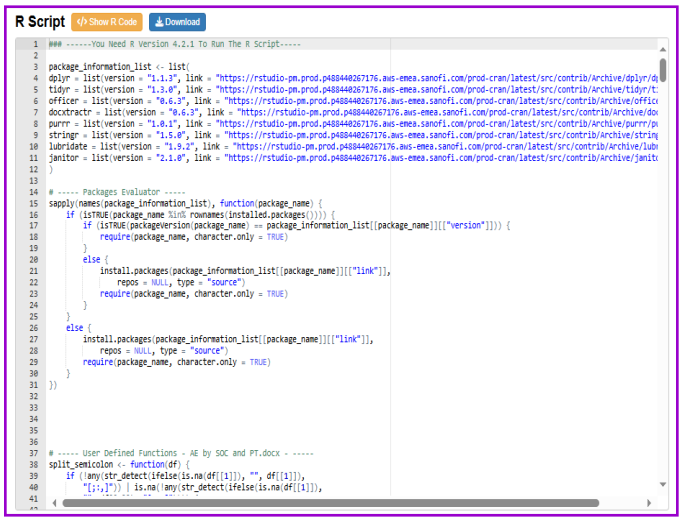

Fig.7: Code Generation

Graph DesigneR:

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually. We developed R shiny application named Graph DesigneR, which provides a user-friendly interface for graph generation. User can import downloaded data into the application and create graph using preloaded libraries or from scratch (Fig. 8). Finally, graph can be published, width, height, aspect ratio can be adjusted, and plots can be saved in different formats.

Graph DesigneR is a large wrapper around the R package **ggplot2**, which is a system for declaratively creating graphs, based on **"The Grammar of Graphics".** Graph DesigneR is also regenerative. Respective code is generated along with graphs, which covers all the process from data input, data manipulation to visualization. We have presented and published about the Graph DesigneR at PHUSE US Connect 2023 (PRE_DV06.pdf, PAP_DV06.pdf).

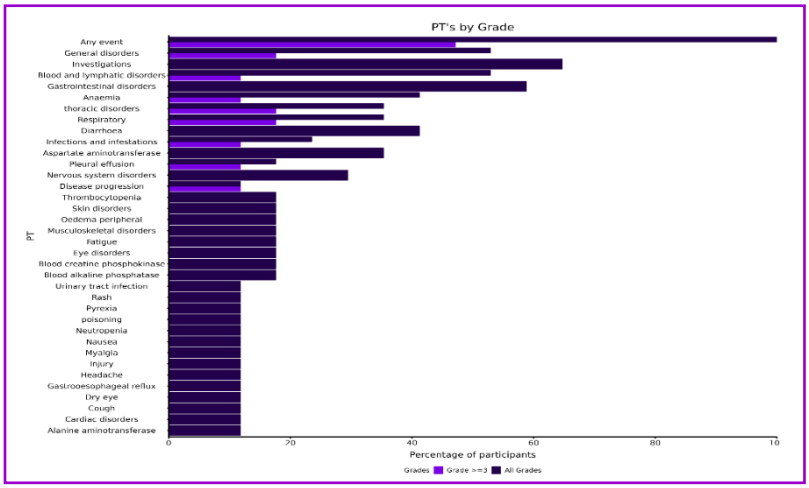

Fig.8: Graphical Visualization

CONCLUSION:

The **Data ExtractoR** platform is a comprehensive and user-friendly tool for data extraction and transformation, catering to users without prior programming knowledge. It supports importing reports/summary tables in multiple formats (PDF/RTF/DOCX) and provides a seamless workflow for cleaning, manipulating, and reshaping data. Users can transform rows into headers, create custom category columns, clean datasets through automated and manual processes, and reshape data between long and wide formats with ease.

The application leverages powerful R packages like ***dplyr***, ***stringr***, and ***tidyr*** for efficient data processing, and all transformations are accompanied by generated R scripts for transparency and reproducibility. The cleaned and transformed datasets, along with their corresponding R code, can be exported in various formats. The tool is designed to simplify data workflows, making it ideal for preparing data for visualization or further analysis.

REFERENCES:

1. https://dplyr.tidyverse.org/
2. https://camelot-py.readthedocs.io/en/master/
3. https://cran.r-project.org/web/packages/striprtf/striprtf.pdf
4. https://github.com/hrbrmstr/docxtractr
5. https://davidgohel.github.io/officer/
6. https://ipub.com/dev-corner/apps/r-package-downloads/
7. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2023/Connect/US/Florida/PAP_DV06.pdf
8. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2023/Connect/US/Florida/PRE_DV06.pdf

ACKNOWLEDGMENTS:

We gratefully acknowledge our colleagues Renit Anthony, Sanjeev Rathore for **Data ExtractoR** application and Sujeet Kumar, Pricy Jain, Babit Pradhan, Bharath Kumar, Likhith Raju and Shivangi Dixit for **Graph DesigneR** application development. We are also thankful to Advance Visualization Task Force team members for their insightful comments and suggestions in the development of these tools.

CONTACT INFORMATION:

Mukul K Mittal
Sanofi, USA
55 Corporate Drive
Bridgewater, NJ, 08807
Email: mukul.mittal@sanofi.com

Andre Couturier
Sanofi, USA
55 Corporate Drive
Bridgewater, NJ, 08807
Email: andre.couturier@sanofi.com