Paper SD06

Data Designer: Empowering Data Preparation, Transformation, ARD Creation and Summarization for Insights and Decision Making.

Mukul Mittal, Sanofi, Morristown, NJ, USA

Sanjeev Rathore, Quantzig, Bengaluru, India

Andre Couturier, Sanofi, Morristown, NJ, USA

ABSTRACT:

Data Designer is intuitive **R Shiny® application,** built to streamline and accelerate data transformation, as well as to summarize trials data. The tool supports **seamless ingestion** of datasets from various file formats e.g., **SAS**, **XPT**, **CSV**, **Excel**, **Parquet**, and **JSON**.

At its core, Data Designer empowers users to create **Analysis Results Datasets (ARDs)** using *{card}* package by leveraging analysis results Metadata specs. These ARDs can be further summarized using *{gtsummary}* to generate publication ready clinical tables. Application also offers support for multilingual outputs and includes features for generating **mock tables** from dummy data as well.

For customized reporting, users can create summary outputs from the ADaM datasets using *{rtables}*, either through predefined templates or custom logic, with outputs exportable in RTF and DOCX formats. Titles and footnotes are dynamically populated from reflist to ensure consistency/compliance, and R code is generated to support reproducibility and traceability.

INTRODUCTION:

Data import, cleaning, wrangling, and transformation are foundational steps of clinical trials data flow, ensuring that collected information is accurate, consistent, reliable and ready to generate insights that support safety profiling and efficacy evaluation for operational decision making and regulatory submissions. Together, these steps create a robust process that turns complex clinical information into simplified reliable evidence.

As we all know, clinical trial data analysis is governed by regulatory standards and internal reporting conventions. To support the generation of Analysis-ready-datasets (ARDs) and Tables-listing-figures (TLFs), Sanofi has recently adopted new standards: the CDISC ARS guidelines.

To support our statistical programming teams with this transition ART (Analysis and Reporting Technologies) team has developed Data Designer; a **R Shiny®** based application that streamlines data ingestion, transformation, and reporting. The application enables seamless handling of industry-standard data formats, including SAS, XPT, CSV, Excel, Parquet, and JSON. The data can be filtered on multiple conditions, new variables can be derived, and missing values can be imputed using multiple methods. Datasets can be merged according to different methods using multiple key variables. Furthermore, the application generates R code, ensuring traceability and reproducibility.

Currently, the application supports ARD generation for key domains such as Demographics, Medical History, and Adverse Events using predefined TLF configurations. These configurations automatically drive variable mapping, analysis specifications, and display attributes. Users can generate stratified summaries, frequency-based adverse event tables, header counts, overall columns, and “Any Event” rows consistent with ARS (Analysis results specifications). Clinical tables can be produced using {***gtsummary)*** R package, either directly from ARDs or through ARS-aligned templates. Outputs are generated with titles, footnotes, and orientations, populated from Analysis Output Metadata (AOM) and reflist metadata, ensuring consistency across all deliverables.

Technically, Data Designer is developed using a modular R architecture built on the Shiny framework. Core dataset derivation and manipulation rely on {***dplyr}*** package functions, including *select (), mutate (), filter (), arrange (),* and *summarize ()*. Reporting outputs leverage {***cards}*** and {***gtsummary}*** packages to meet both standard and custom reporting needs. The application generates fully reproducible R scripts, including standardized header capturing tool i.e., version, author, and study information. This technical foundation enables Data Designer to deliver ARS-compliant, reproducible, and submission-ready clinical outputs.

METHODOLOGY AND RESULTS:

MULTI-FORMAT DATA SUPPORT:

Data analysis begins with the essential step of importing data into the application. To create a comprehensive master dataset, users often require multiple raw datasets. Data Designer is specifically designed to support this requirement. Users can seamlessly import multiple datasets of different formats through an intuitive interface (Fig. 1).

To ensure efficient handling of various file formats, different R packages are used, including *{readr}* for CSV files, *{openxlsx}* for Excel files, *{haven}* for SAS and XPT files, *{jsonlite}* for JSON files, and {*arrow}* for Parquet files. This tailored approach enhances the application's versatility and enables users to work with diverse datasets efficiently.

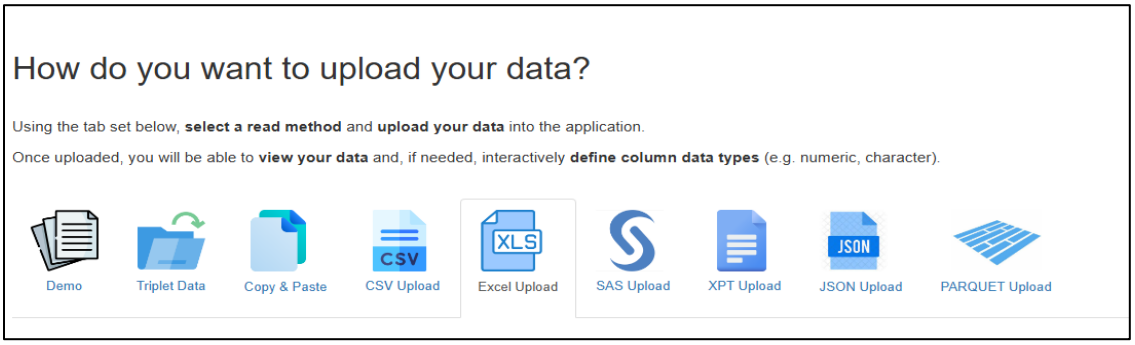

Figure 1: Multi-Format data Ingestion

ADVANCE DATA SUBSETTING:

Enhanced filtering functionality allows users to define multiple filters, each composed of one or more logical conditions, enabling complex subsetting scenarios without manual programming. In addition, it also enables value comparisons between columns, supporting flexible dataset subsetting based on inter-variable dependencies (e.g. TRTSDT > TRTEDT). Specialized functions for **POSIXct and POSIXlt classes** enable accurate filtering of *date and date-time variables*. These capabilities support controlled and reproducible subsetting. We have used different functions from *{dplyr}* package like *select ()* and *filter ()* to select the variables and filter the data based on one or multiple conditions(s).

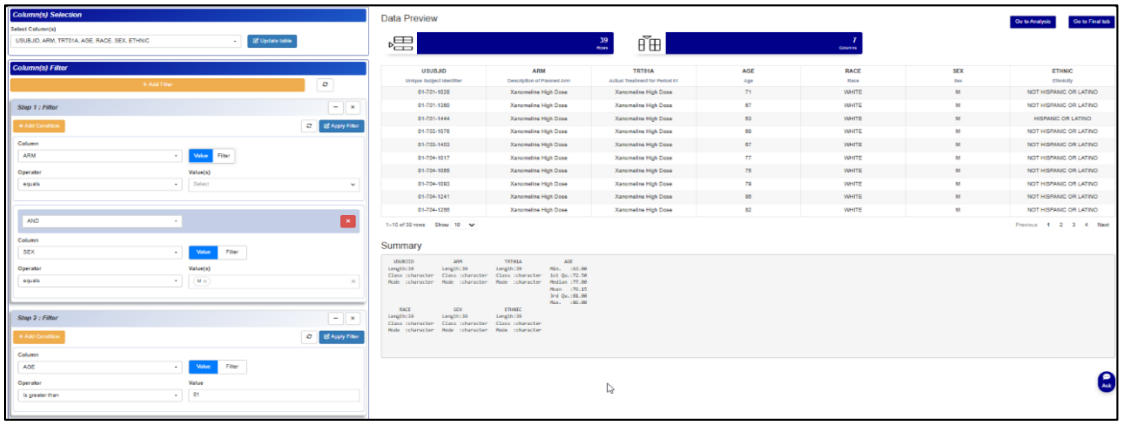

Figure 2: Advanced Data Subsetting

MISSING VALUE IMPUTATION:

In clinical trials, collected data may have missing value due to patient dropout, measurement errors or absence at some time point for some subjects. There are many methods to deal with missing data. In this application, we utilized predefined imputation strategies e.g., mean, median, mode, KNN (K Nearest Neighbors), and last/first observation carried forward, or by removing incomplete observations when appropriate. It reduces the need for manual intervention during data preparation.

Missing value imputation is implemented using various packages and established methods, e.g., *mean ()* and *median ()* functions from *{base R}*, *mfv ()* function from *{modeest}* for mode-based imputation, *KNN ()* from *{VIM}*, and *na.locf ()* from *{zoo}* for first/last-observation-carried-forward approaches.

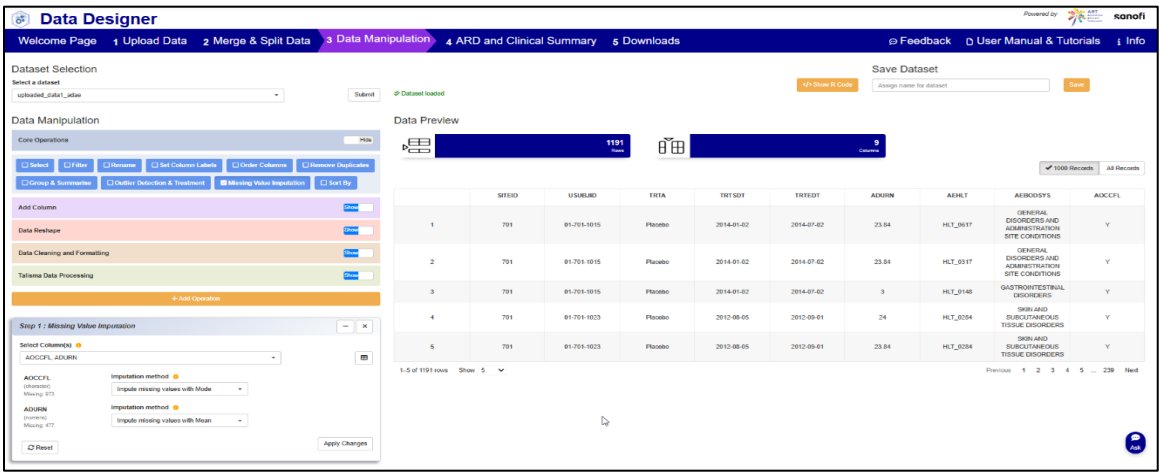

Figure 3: Imputation for missing values

MERGING, APPENDING and SPLITTING DATASET(S):

Application provides comprehensive dataset merging and split functionality to support flexible data preparation. Datasets can be merged either by columns (join operations) or by rows (append/bind functionality) and can also be split based on *row ranges*. Column-wise merging supports multiple join types e.g., left join, right join, inner join, full join, anti-join, and semi-join. When appending datasets by rows, records with common column names and data types are combined, while *non-matching columns* result in the creation of new columns populated only from the respective dataset. *Anti and semi joins* are specifically designed for record filtering rather than full column merging e.g., an anti-join can be used to identify subjects without adverse events by merging ADSL dataset with ADAE by USUBJID.

From an implementation perspective, various functions are used from *{dplyr}* package to perform all column-wise merge operations, including *inner_join (), left_join (), right_join (), full_join (), anti_join (), and semi_join ()*. Row-wise merging is implemented using *bind_rows ()*, while dataset splitting and subsetting are achieved through *slice ()* and *select ()* functions.

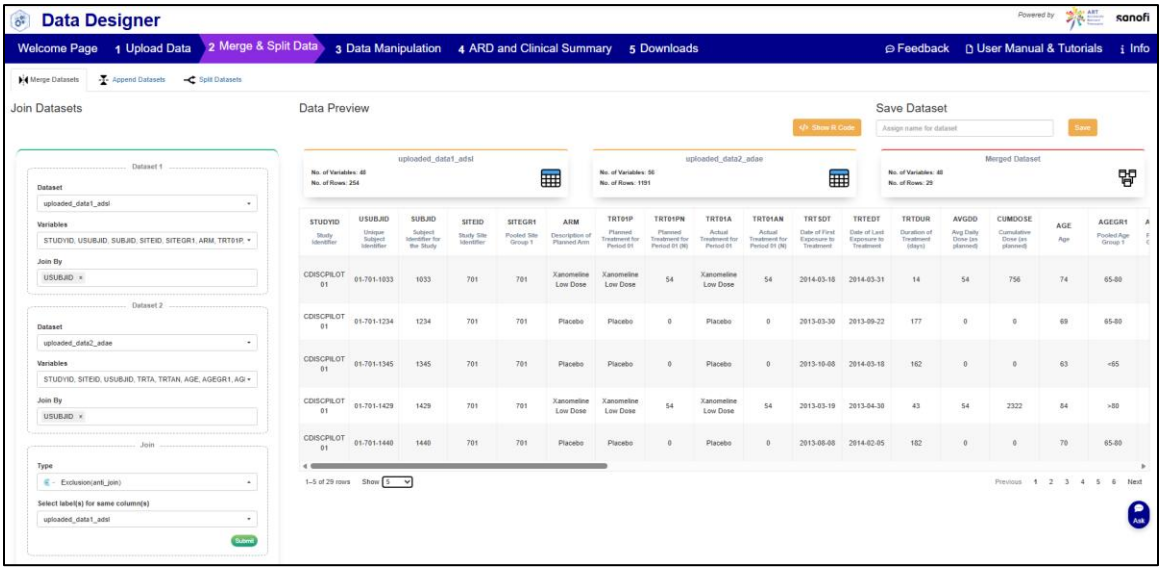

Figure 4: Dataset joining using anti-join finction

GROUPING AND SUMMARIZATION:

The **Group and Summarize** feature provide a flexible framework for aggregating dataset based on user-defined grouping variables, summarize variables, and selected aggregation functions. For continuous variables, it offers a range of aggregate functions, including *sum, mean, minimum, maximum, median, standard deviation, standard error, and quantiles*, enabling comprehensive statistical summarization. For categorical variables, *counts* are computed to summarize frequency distributions. Users can further refine output by specifying the number of decimal points for the aggregated values, ensuring consistent numerical presentation. Additionally, the feature allows custom renaming of summarized columns, providing clarity and enhancing interpretability of the results.

From a technical perspective, this functionality is implemented using the *{dplyr}* package, with core operations driven by functions such as *group_by (), summarise (), and rename ()*. Aggregation computations leverage standard *{base R}* functions, including *mean (), min (), max (), n () etc.*, ensuring efficient and reproducible summary generation.

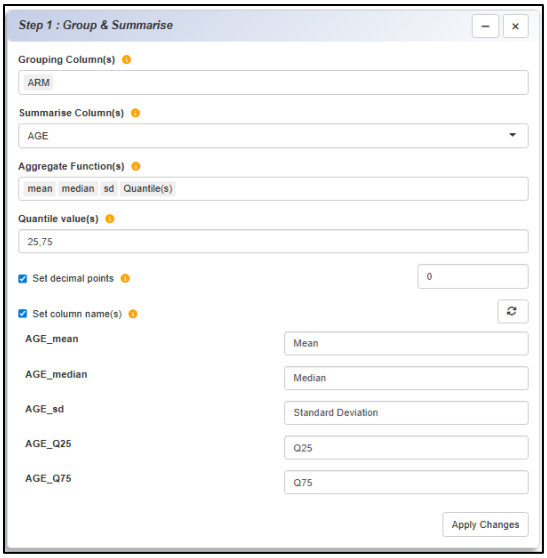

Figure 5.1: Input configuration panel

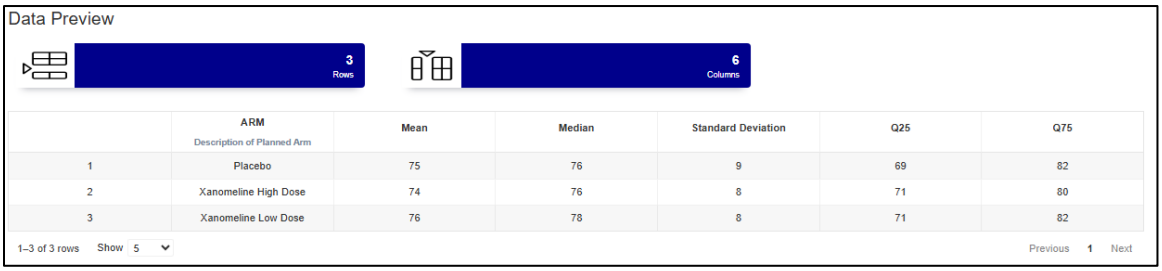

Figure 5.2: Summarized output

ARD AND SUMMARY CREATION USING SANOFI ARS:

ARD creation within the *ARD & Summary (Sanofi ARS)* module is driven by Sanofi ARS–compliant TLF configurations. The application automatically loads the selected TLF specifications and applies it to generate standardized ARD. *Automatic variable mapping* is performed based on TLF specs, with the flexibility to apply manual mapping for unmapped variables. ARD generation is currently supported for some tables e.g., Demographics, Medical History, and Adverse Events.

We have extensively used the complex list structure to capture the configuration/meta data about ARS as well as *ard_stack ()*, *ard_stack_hierarchical ()* functions from *{cards}* package to generate the ARD.

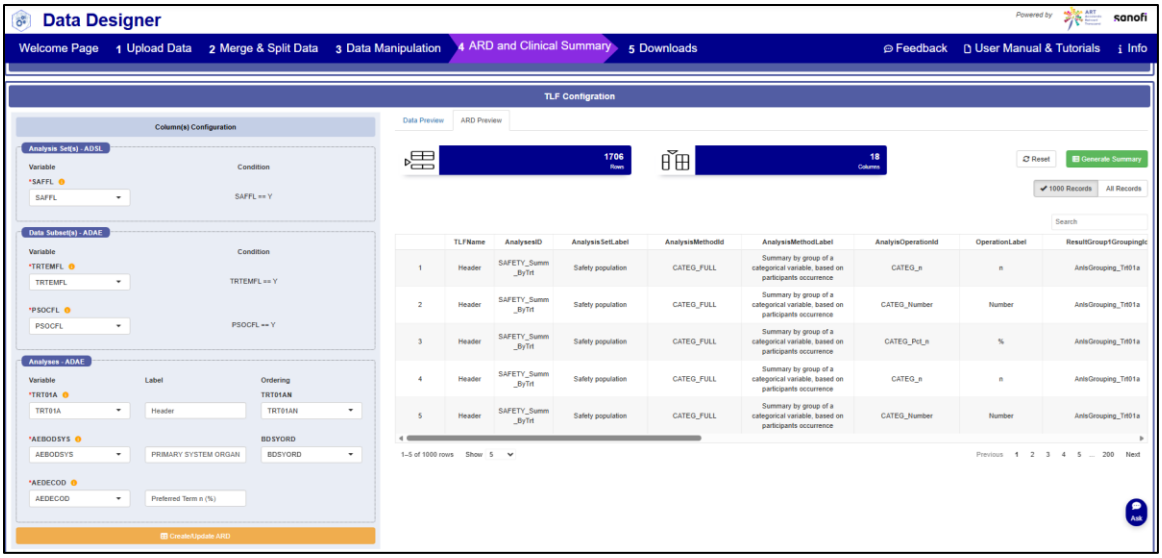

Figure 6.1: ARD Creation

Summary creation is tightly integrated with ARD, enabling one-click summary output directly from the generated ARD. The module applies ARS-defined analysis logic and sorting rules to produce compliant clinical tables. *Display attributes* such as titles, footnotes, and table orientation are automatically populated from AOM, ensuring consistency with approved reporting standards. Users can download summaries in DOCX and HTML formats. This seamless transition from ARD to summary outputs significantly reduces time and effort.

At the backend, we are using several functions from *{gtsummary}* package like *tbl_ard_summary (), tbl_ard_hierarchical (), modify_table_styling (), modify_header ()* etc. to generate summary reports.

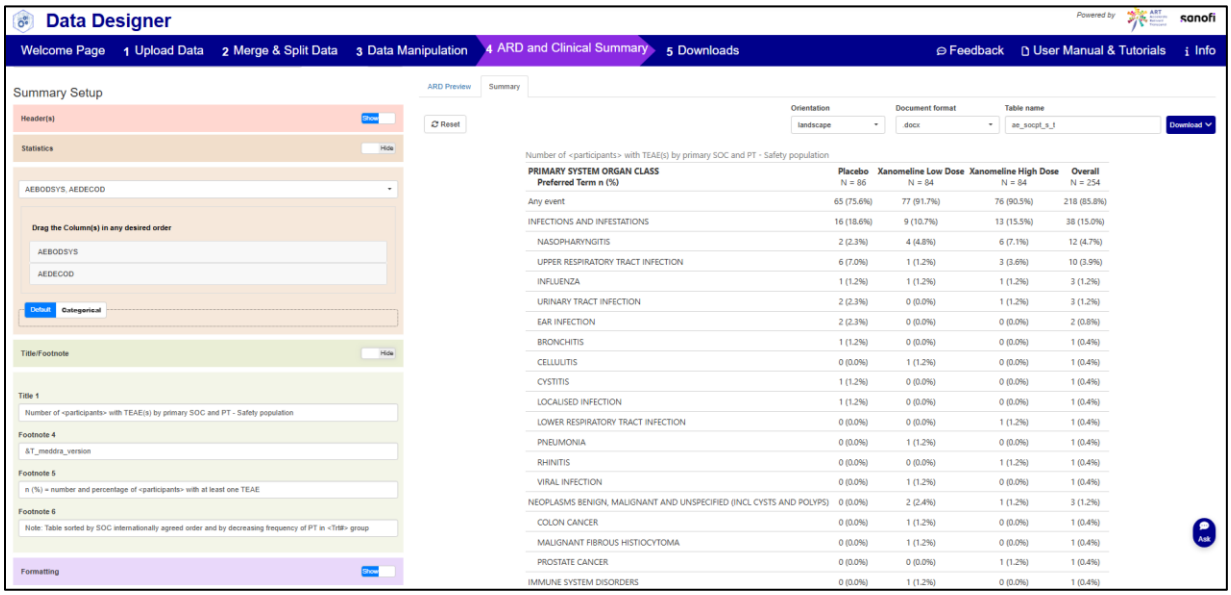

Figure 6.2: Summary creation using ARD

Overall, the ARD & Summary outputs functionality provides an **end-to-end, metadata-driven workflow** for standardized clinical reporting. By integrating TLF configuration, variable mapping, ARD creation, and summary generation within a single module, ensures traceability and reproducibility. Automation minimizes variability while preserving flexibility for study-specific needs. Collectively, this capability strengthens analytical consistency and operational efficiency across clinical studies.

MOCK SUMMARY CREATION WITH MULTILINGUAL SUPPORT:

The mock summary creation allows users to generate mock shells / clinical summaries without revealing actual data. By using this feature, the structure and layout of summary tables can be shared with all team members while maintaining data confidentiality. When creating ARDs with custom logic, users can enable **"Convert to Mock ARD"** option and subsequently generate summaries from the resulting dataset. The resulting summaries display placeholder values such as "xx" instead of real aggregated numbers, ensuring sensitive data is protected. This capability facilitates review and validation of summary tables prior to final data availability. Moreover, the tool supports multi-language summary generation, enabling tables to be presented in languages such as *Chinese* or *Japanese*. Language selection is configurable at the time of summary creation, supporting *global collaboration*.

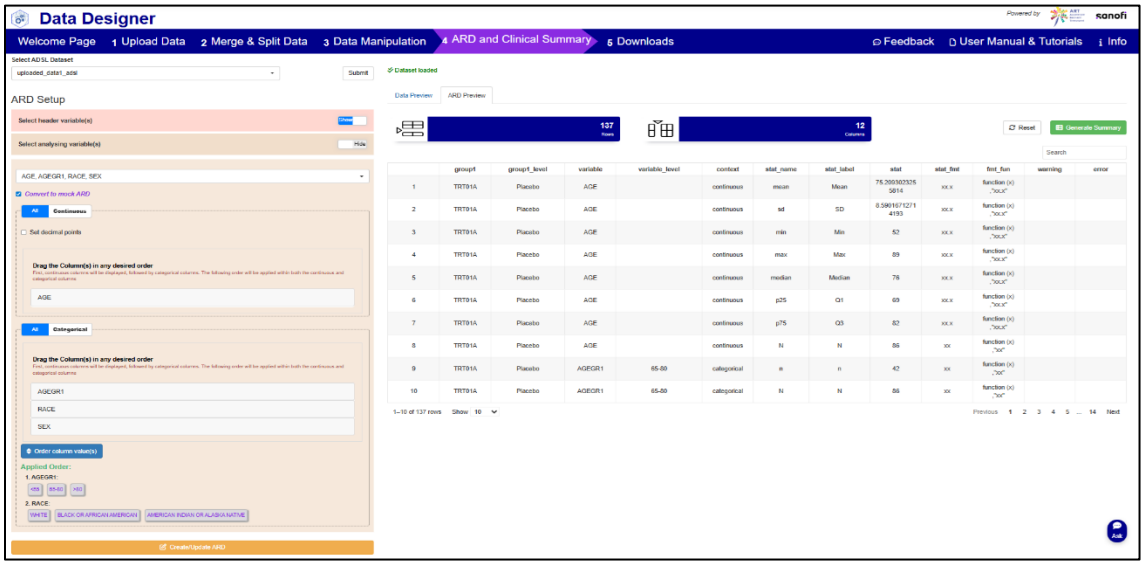

Figure 7.1: Creating mock ARD dataset using custom logic

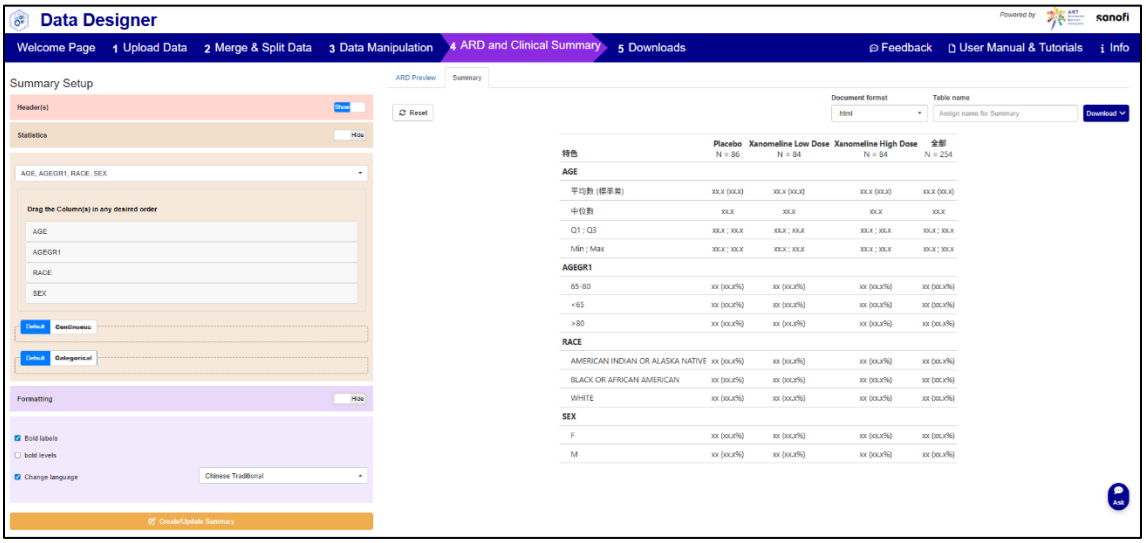

Figure 7.2: Creating mock summary

TEMPLATE-BASED SUMMARY GENERATION:

Application also enables the creation of clinical summaries using *{rtables} framework* from **ADaM datasets**, leveraging a **set of pre-defined, standardized templates**. The currently available templates include *Demographics* and *Adverse Event* tables. Each template includes an *information panel* detailing essential metadata, such as the required dataset, mandatory variables, and optional variables. Upon template selection, a configuration form is launched, enabling users to define *reporting parameters* and generate summaries based on the *configured variables*. Users can further map the title/footnote using the *reference list (reflist)* from a specific study to ensure alignment with study metadata and standards. The generated output can be exported in multiple formats, including HTML, RTF, PDF, and DOCX.

From a technical perspective, the implementation is built using the *{rtables}* package, with core table generation driven by key functions such as *basic_table (), summarize_row_groups (), and split_cols_by ()*. Additional layout enhancements, including the creation of an overall summary column, are achieved using *add_overall_col ()* function.

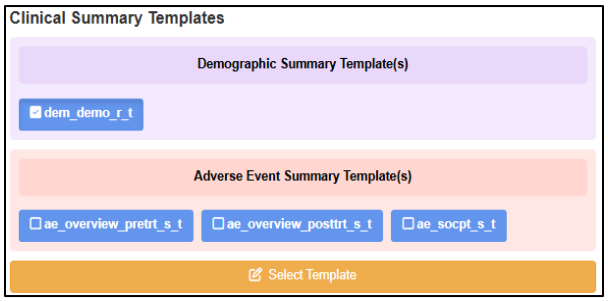

Figure 8.1: Template selection panel

Figure 8.2: Template information panel

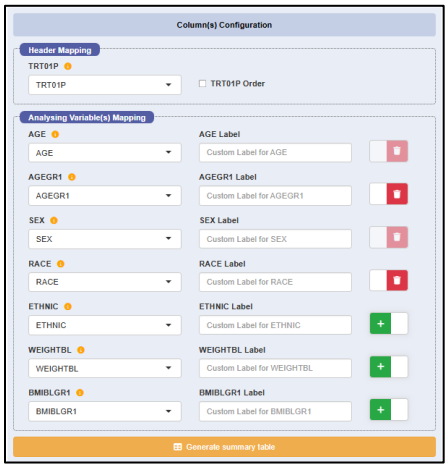

Figure 8.3: Configuration panel

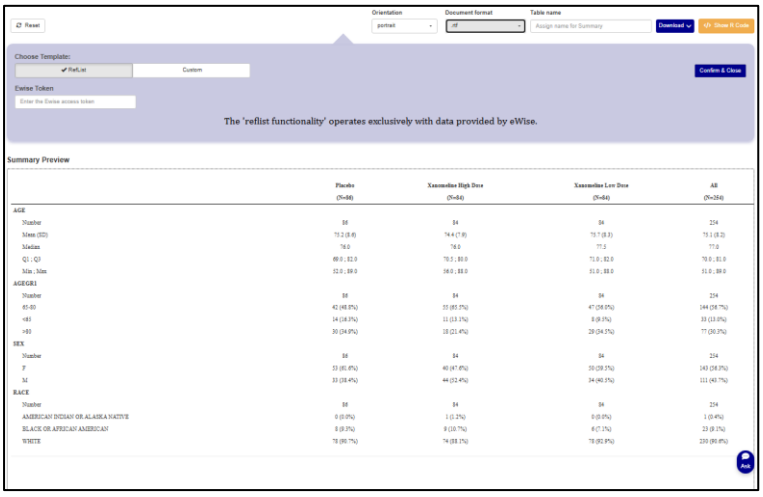

Figure 8.4: Reference list mapping and summary output

CONCLUSION:

Data Designer is an intuitive, user-friendly R Shiny® application that streamlines clinical data transformation, analysis and summary generation. Through its interactive interface, users can manipulate, map, and summarize data without extensive programming, while leveraging a robust R framework built on *{dplyr}, {cards}, and {gtsummary}* R packages.

The tool supports simultaneous processing of multiple input datasets from diverse formats, including CSV, Excel, SAS, XPT, JSON and Parquet. Generated datasets and summaries can be exported in multiple formats, enabling seamless integration with downstream visualization or reporting workflows. Data Designer also facilitates the creation of clinical tables directly from analysis-ready datasets (ARDs).

A key feature of Data Designer is its regenerative capability: reproducible R code is automatically generated alongside each data processing and reporting step. This promotes reusability, validation, and traceability to ensure reproducibility, and maximizes generalizability.

Future enhancements will focus on expanding support for ARS-based summary creation, enabling the generation of a final report that includes all possible summaries in a single document, as well as supporting the creation of multiple summaries in one go.

REFERENCE:

1. https://dplyr.tidyverse.org/
2. https://insightsengineering.github.io/cards/main/index.html
3. https://www.danieldsjoberg.com/gtsummary/index.html
4. https://www.rdocumentation.org/packages/jsonlite/versions/2.0.0
5. https://arrow.apache.org/docs/r/

ACKNOWLEDGMENTS:

We gratefully acknowledge our colleagues Parth Patadia (Quantzig), Vatsal Garg (Quantzig), Ramya Pichikala (Sanofi) and Sophie Bellec (Sanofi) for their contribution and suggestions to the development of this tool.

CONTACT INFORMATION:

Mukul K Mittal
Sanofi, USA
100 Morris St,
Morristown, NJ, 07960
Email: mukul.mittal@sanofi.com

Andre Couturier
Sanofi, USA
100 Morris St,
Morristown, NJ, 07960
Email: andre.couturier@sanofi.com