PharmaSUG 2023 - Paper SD-230

Automated Mockup Table and Metadata Generator

Jeff Cheng, Shunbing Zhao, Guowei Wu, and Suhas R. Sanjee, Merck & Co., Inc., Rahway, NJ, USA

ABSRACT

Preparation and production of an analysis and reporting (A&R) package for a regulatory filing is a resource intensive process. The A&R package includes and is not limited to analysis datasets, tables, listings, and figures (TLFs). To ensure that the TLFs meet the needs of the stakeholders, mockup TLFs are generated by the statisticians to communicate the TLF specifications to the statistical programming team. Currently the mockup TLF generation process is a manual and time-consuming process. In addition, a statistical programmer creates call programs and the related documentation based on the mockups. Populating and maintaining the same information consistently in multiple documents is challenging and tedious. In order to streamline this process, we developed a proof-of-concept R Shiny app that provides a user-interface to: 1.) gather and store metadata required for generating TLFs once and use it to populate all relevant documents 2.) automatically generate mockup TLFs, required metadata and draft call programs for standard macros based on user inputs 3.) make it easy to reuse and adopt mockup specifications from an existing study for a new one 4.) create a preview of mockup TLFs and related documents. In this paper, we will describe the Shiny app, discuss what we have learned and the next steps.

INTRODUCTION

To prepare for A&R package generation for a regulatory submission, the TLF specifications including but not limited to the types of TLF, information about the source data, and output file names are defined by the statistician in a spreadsheet. This spreadsheet is referred to as "A&R Grid" in the rest of this paper. In addition, the statistician creates mock table shells with more information on the content and layout of the TLFs. Using the information from the study protocol, statistical analysis plan (SAP), A&R grid and mockup table shells, the statistical programming team then develops programs to generate TLFs. Some TLFs are generated using the global /therapeutic standard macros and the rest are generated using protocol-specific macros. The statistical programmers refer to the standard macros' user manuals and create programs to call the standard macros to generate the standard TLFs. These TLFs are then reviewed by the study statistician and statistical programming team to make sure that they are accurate.

With hundreds of TLFs included in a package, each with dozens of parameters that need to be specified, it is tedious to customize standard mockups with study-specific information using the aforementioned process: Generation of the A&R grid and call program are time-consuming and resource-intensive. Mismatches between the A&R grid and mockup TLFs are hard to catch. The same parameter sometimes needs to be updated in multiple files. As a result, the quality of a filing package can be negatively impacted by the increase in the number of TLF required by a filing package using existing process.

As outlined in the paper that our team published in 2020 [1], we explored extracting information from the study documents – study protocol, SAP and mock table shells using Natural Language Processing (NLP) to extract the metadata and use it to automatically create programs to call standard TLF macros. Based on the feasibility analysis, we realized that extracting metadata from study documents is a non-trivial problem, owing to several challenges, given that study documents have a lot of information in free text and that NLP has limitation processing semantics.

With these pain points in mind, we embarked on prototyping an interactive application with a graphical user interface (GUI) front-end to allow statisticians to define the TLF mocks and to generate metadata in a structured format. We used some elements from the TFL Automation Engine proof of concept (POC) design published by CDISC 360 working group[2], but this is customized to our internal A&R framework. This application aims to integrate the processes of customizing TLF parameters based on study-specific information with the creation of template programs required for generating TLFs using SAS. Additionally, it includes the production and preview of corresponding mock TLFs to make sure they meet expectations of the statisticians. In addition, values for TLF parameters and study-specific info can be saved and reloaded for adaptation of A&R package production of a future study. The extent to which the process of A&R package production can be streamlined and tracked with improved quality are being assessed at the completion of this prototype development.

METHODOLOGY & MATERIALS

Because of its large library of interactive widgets, relative ease of use, and active and robust online support and learning resources, R Shiny was selected as the programming platform to build the web-based Automated Mockup Table and Metadata Generator. The application is hosted by a Posit Connect server, which allows multiple users to launch the application using web browser simultaneously.

Functions from the following R packages were used in implementing the application:

shiny: Web application framework for R

shinyjs: Improve user experience of Shiny apps with JavaScript

shinyfiles: A server-side file system viewer for Shiny

markdown: Render markdown with the C library 'sundown'

shinythemes: Themes for Shiny

reactable: Interactive data tables based on 'React Table'

reactablefmtr: Easily customize interactive tables made with Reactable

mass: Support functions and datasets for Venables and Ripley's mass

shinymatrix: Shiny matrix input field

shinybs: Twitter bootstrap components for Shiny

dt: A wrapper of the JavaScript library 'datatables'

sortable: Drag-and-drop in 'Shiny' apps with 'sortablejs'

tidyverse: Easily install and load the 'tidyverse'

magrittr: A forward-pipe operator for R

r2rtf: Easily create production-ready rich text format (RTF) table and figure

openxlsx: Read, write and edit xlsx files

stringr: Simple, consistent wrappers for common string operations

mksurv: BARDS internal standard R package for survival analysis

Information from existing A&R grid documents, standard macros, metadata and their associated TLFs were used as references during development. Statisticians and statistical programmers with regulatory filing and R programming experiences were recruited to be members of the development team.

HIGH LEVEL DESIGN

The overall workflow and design of the application is depicted in Figure 1 below: A&R metadata and documentation containing information about the TLFs such as analysis populations, analysis variables, sub-setting conditions and other parameters, are specified by a user via the frontend GUI and stored in a structured dataset.

Template call programs, A&R metadata documentation and mock TLFs are then generated by processing engine of the application using information stored in the structured dataset. With minor modifications by the study programmers based on the analysis needs, template call programs and A&R metadata documents can be quickly updated to generate call programs to standard TLF macros and generate the final TLFs. The template call programs, metadata and mock TLFs can be previewed using the application to ensure accuracy.

Finally, user-specified A&R parameters can be saved as an R data object and reloaded for future A&R deliverables.

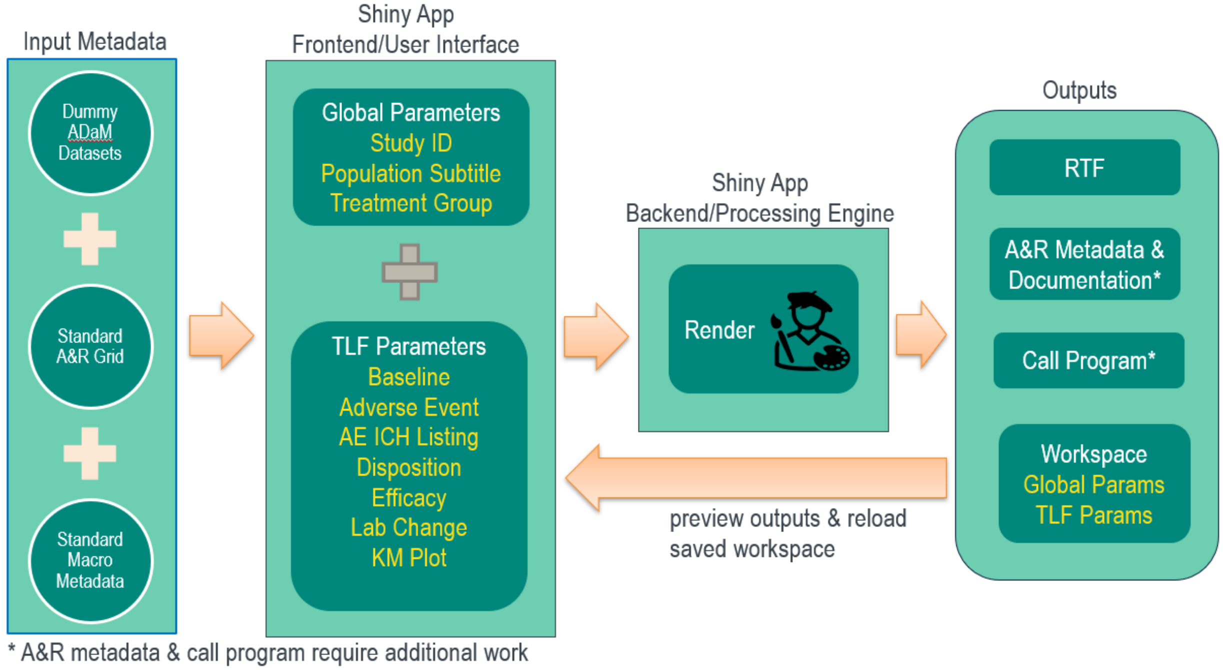
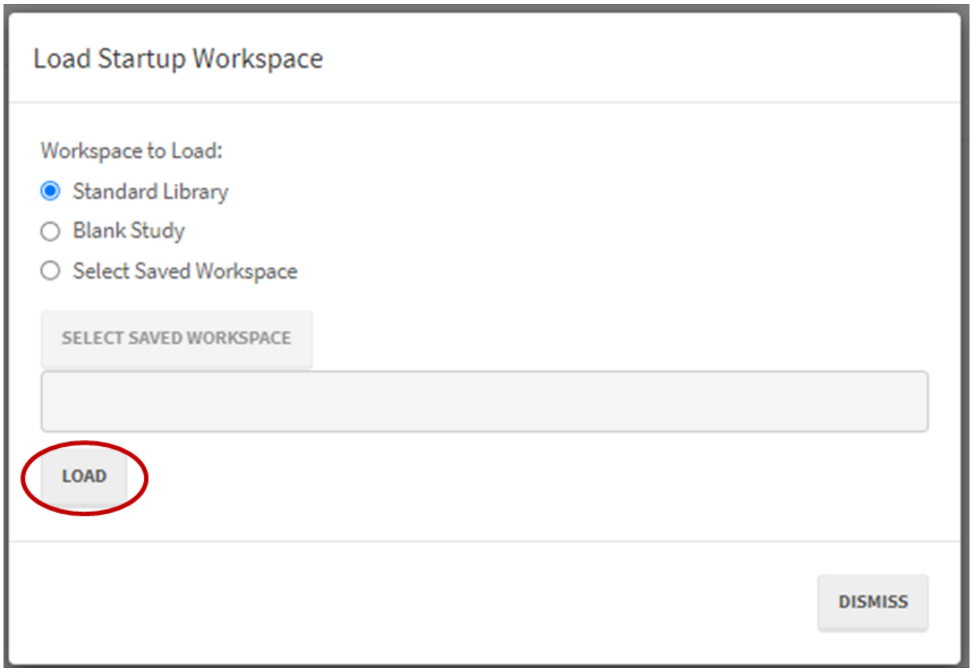


Figure 1. Automated Mockup Table and Metadata Generator workflow and design

In this POC, we chose 7 mockups to build up. As shown in the Load Startup Workspace screen in Figure 2, on startup the user has the choice of starting with either pre-populated standard macro default parameters for all 7 implemented TLFs, a previously saved workspace, or a blank workspace.:



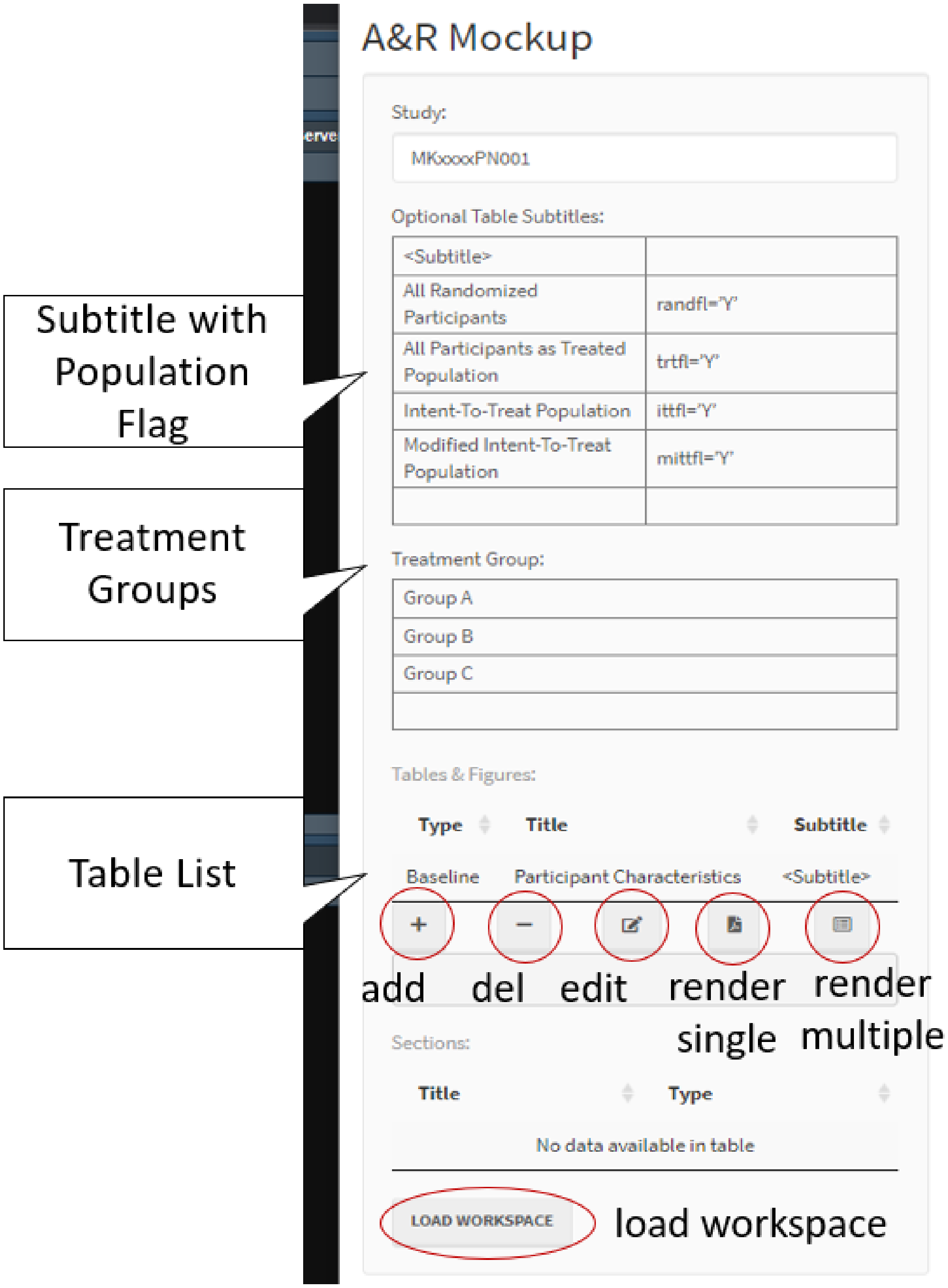
Load Startup Workspace

Workspace to Load:

● Standard Library
○ Blank Study
○ Select Saved Workspace

Figure 2. Select startup workspace

The main page (Figure 3) contains widgets for entering study-specific information and controls for creating/editing analysis populations and treatment group labels. Once these are defined, Figure 4 shows the interface to define parameters for each table.



A&R Mockup

Study:

MKxxxxPN001

Optional Table Subtitles:

<Subtitle>	
All Randomized Participants	randfl='Y'
All Participants as Treated Population	trtfl='Y'
Intent-To-Treat Population	ittfl='Y'
Modified Intent-To-Treat Population	mittfl='Y'

Treatment Group:

Group A
Group B
Group C

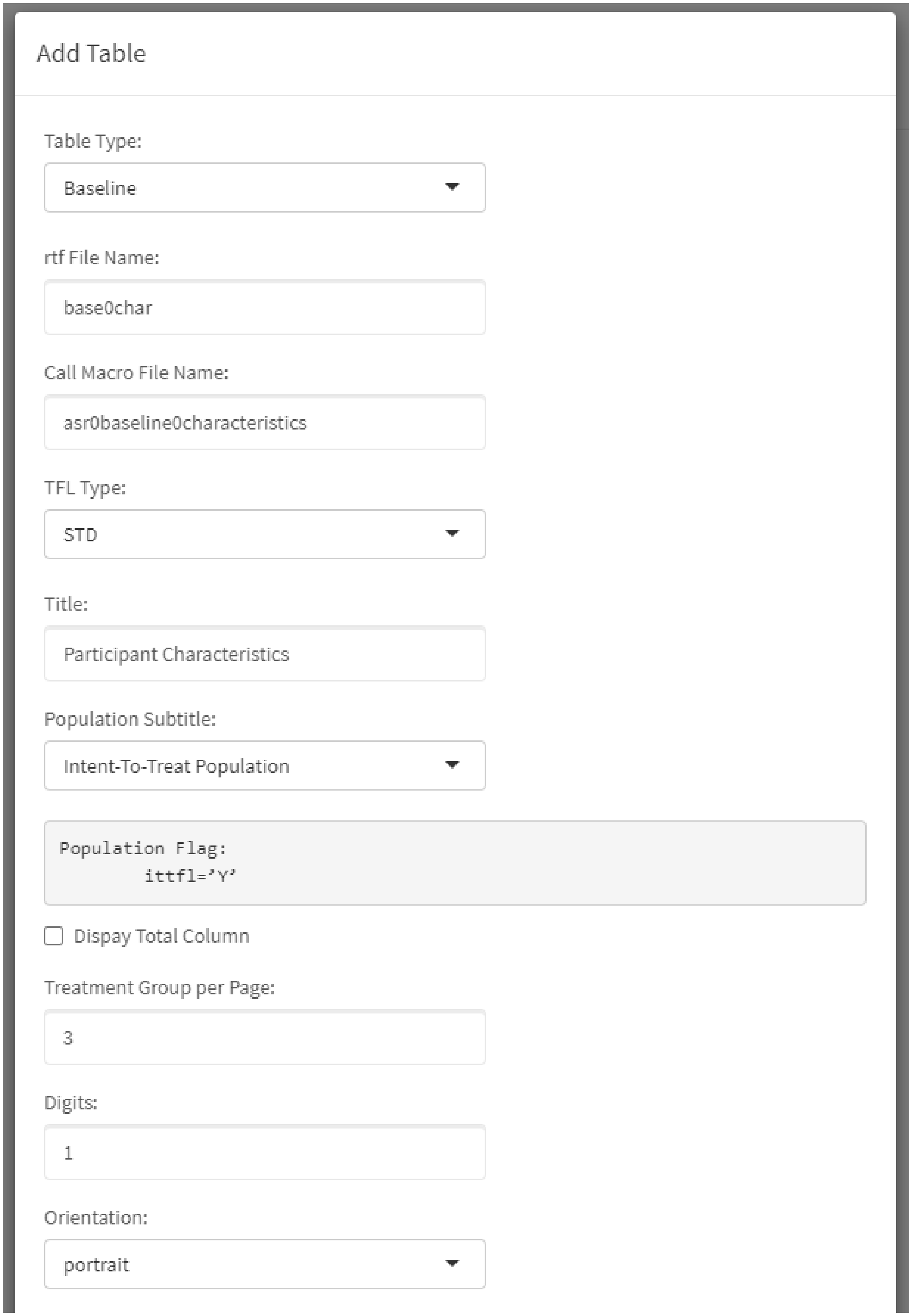
Tables & Figures:

Type	Title	Subtitle
Baseline	Participant Characteristics	<Subtitle>

Sections:

Title	Type
No data available in table	

Figure 3. Widgets for entering study-specific information and controls for creating & editing TLFs. 'render single' button to render currently selected TLF in the table list. 'render multiple' button for selecting multiple TLFs for rendering



Add Table

Table Type:

Baseline ▾

rtf File Name:

base0char

Call Macro File Name:

asr0baseline0characteristics

TFL Type:

STD ▾

Title:

Participant Characteristics

Population Subtitle:

Intent-To-Treat Population ▾

```
Population Flag:
        ittfl=’Y’
```

☐ Dispay Total Column

Treatment Group per Page:

3

Digits:

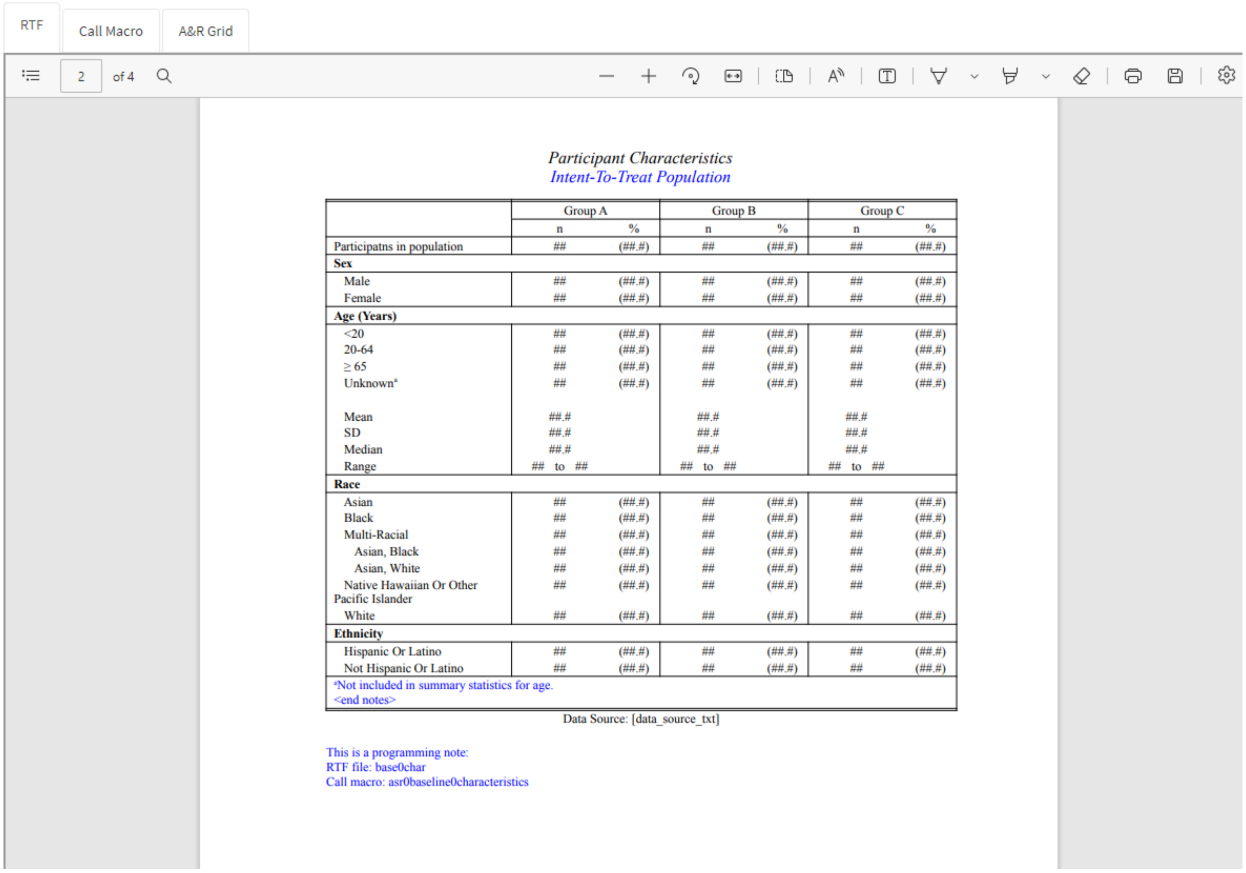
1

Orientation:

portrait ▾

Figure 4. customize parameter settings for individual TLF

After customizing all parameters, the application will create corresponding mock TLFs in a single RTF with a hyperlinked table of contents, template grid containing A&R metadata, and template SAS call programs for standard macros. One of the output mockup table document pages is shown in Figure 5. The users have an option to preview the outputs and make changes as needed.



Participant Characteristics
Intent-To-Treat Population

	Group A		Group B		Group C	
	n	%	n	%	n	%
Participatns in population	##	(##.#)	##	(##.#)	##	(##.#)
Sex						
Male	##	(##.#)	##	(##.#)	##	(##.#)
Female	##	(##.#)	##	(##.#)	##	(##.#)
Age (Years)						
<20	##	(##.#)	##	(##.#)	##	(##.#)
20-64	##	(##.#)	##	(##.#)	##	(##.#)
≥ 65	##	(##.#)	##	(##.#)	##	(##.#)
Unknown[a]	##	(##.#)	##	(##.#)	##	(##.#)
Mean	##.#		##.#		##.#	
SD	##.#		##.#		##.#	
Median	##.#		##.#		##.#	
Range	## to ##		## to ##		## to ##	
Race						
Asian	##	(##.#)	##	(##.#)	##	(##.#)
Black	##	(##.#)	##	(##.#)	##	(##.#)
Multi-Racial	##	(##.#)	##	(##.#)	##	(##.#)
Asian, Black	##	(##.#)	##	(##.#)	##	(##.#)
Asian, White	##	(##.#)	##	(##.#)	##	(##.#)
Native Hawaiian Or Other Pacific Islander	##	(##.#)	##	(##.#)	##	(##.#)
White	##	(##.#)	##	(##.#)	##	(##.#)
Ethnicity						
Hispanic Or Latino	##	(##.#)	##	(##.#)	##	(##.#)
Not Hispanic Or Latino	##	(##.#)	##	(##.#)	##	(##.#)

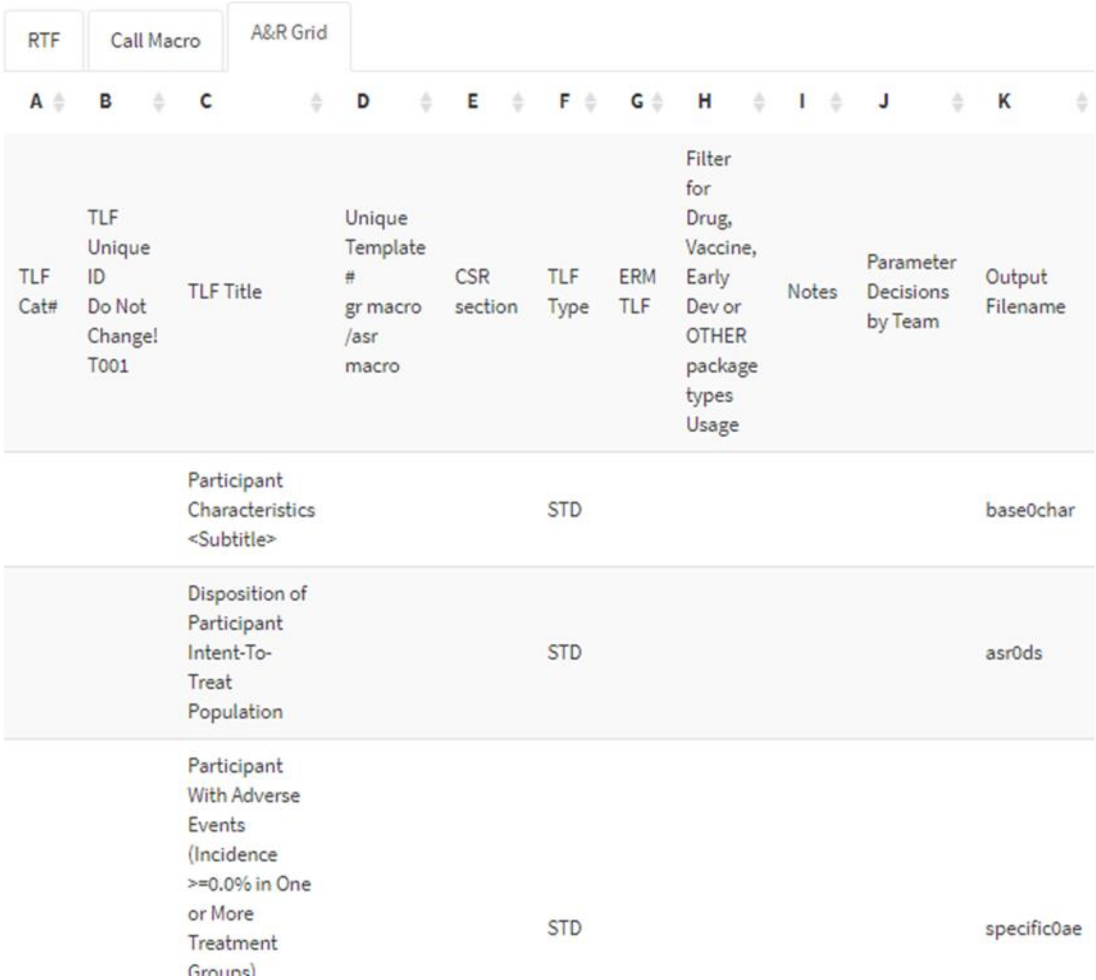
[a] Not included in summary statistics for age.
<end notes>

Data Source: [data_source_txt]

This is a programming note:
RTF file: base0char
Call macro: asr0baseline0characteristics

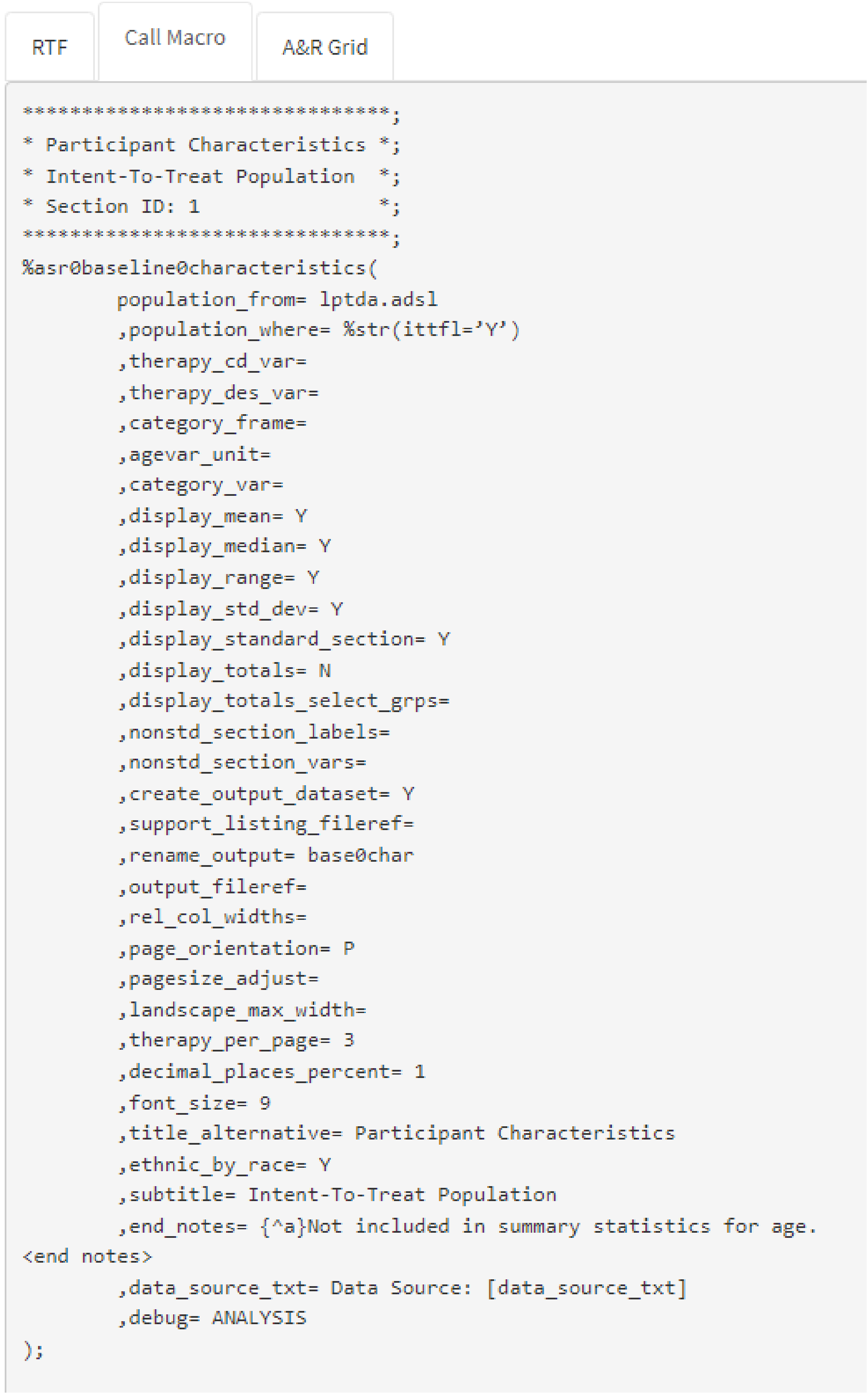
Figure 5. Mock baseline characteristic table

Figure 6 shows the A&R grid populated with metadata. Figure 7 shows the template call program which can be used as starting point for generating the baseline characteristics table.



A	B	C	D	E	F	G	H	I	J	K
TLF Cat#	TLF Unique ID Do Not Change! T001	TLF Title	Unique Template # gr macro /asr macro	CSR section	TLF Type	ERM TLF	Filter for Drug, Vaccine, Early Dev or OTHER package types Usage	Notes	Parameter Decisions by Team	Output Filename
		Participant Characteristics <Subtitle>			STD					base0char
		Disposition of Participant Intent-To-Treat Population			STD					asr0ds
		Participant With Adverse Events (Incidence >=0.0% in One or More Treatment Groups)			STD					specific0ae

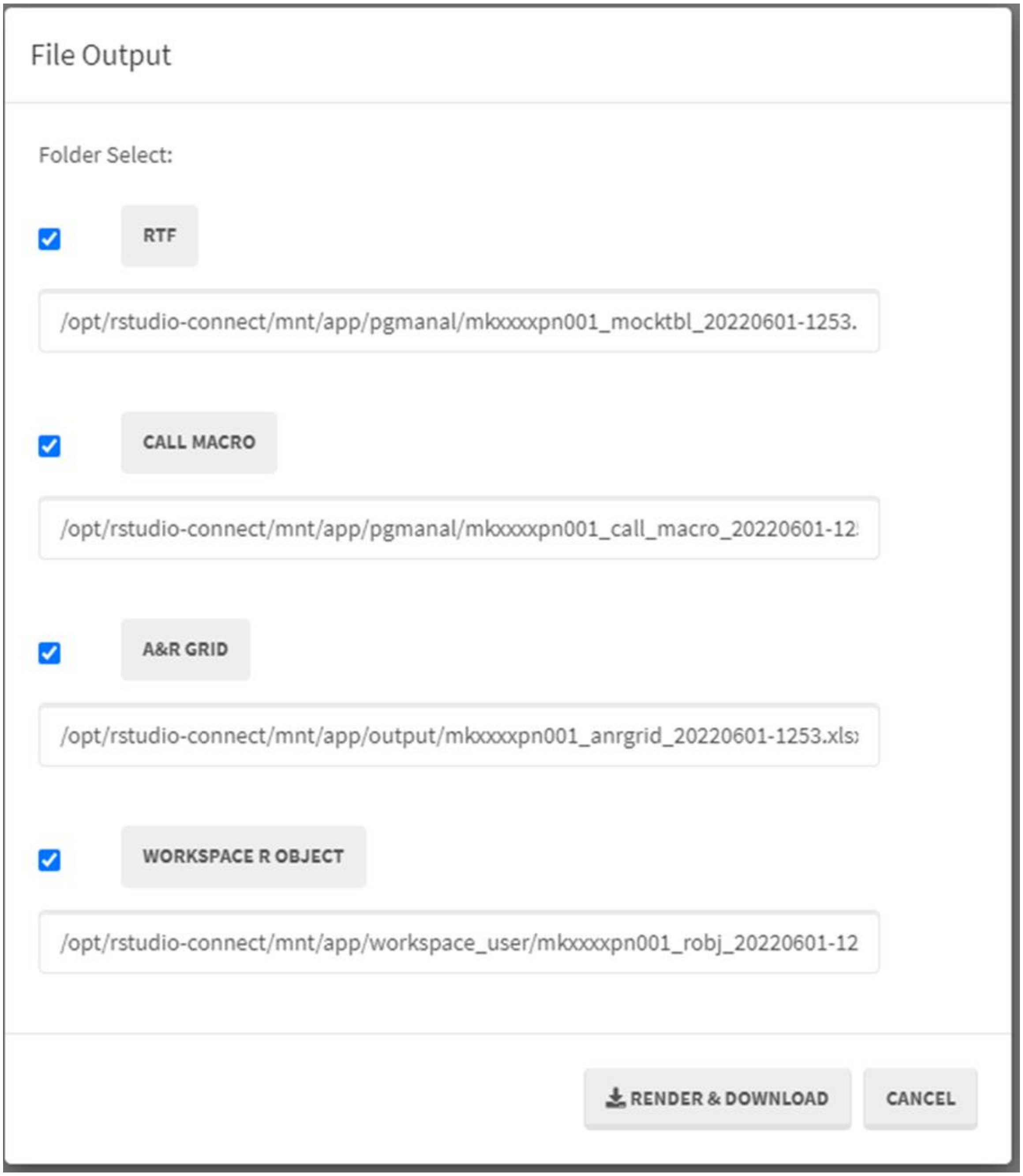
Figure 6. Template A&R grid with metadata



```
*******************************;
* Participant Characteristics *;
* Intent-To-Treat Population  *;
* Section ID: 1               *;
*******************************;
%asr0baseline0characteristics(
        population_from= lptda.adsl
        ,population_where= %str(ittfl=’Y’)
        ,therapy_cd_var=
        ,therapy_des_var=
        ,category_frame=
        ,agevar_unit=
        ,category_var=
        ,display_mean= Y
        ,display_median= Y
        ,display_range= Y
        ,display_std_dev= Y
        ,display_standard_section= Y
        ,display_totals= N
        ,display_totals_select_grps=
        ,nonstd_section_labels=
        ,nonstd_section_vars=
        ,create_output_dataset= Y
        ,support_listing_fileref=
        ,rename_output= base0char
        ,output_fileref=
        ,rel_col_widths=
        ,page_orientation= P
        ,pagesize_adjust=
        ,landscape_max_width=
        ,therapy_per_page= 3
        ,decimal_places_percent= 1
        ,font_size= 9
        ,title_alternative= Participant Characteristics
        ,ethnic_by_race= Y
        ,subtitle= Intent-To-Treat Population
        ,end_notes= {^a}Not included in summary statistics for age.
<end notes>
        ,data_source_txt= Data Source: [data_source_txt]
        ,debug= ANALYSIS
);
```

Figure 7. Template SAS call program

All template documents and mock TLFs can be saved as files for modification and further examination. The parameter setting workspace can be saved and reused, as shown in Figure 8.



File Output

Folder Select:

☑ RTF

/opt/rstudio-connect/mnt/app/pgmanal/mkxxxxpn001_mocktbl_20220601-1253.

☑ CALL MACRO

/opt/rstudio-connect/mnt/app/pgmanal/mkxxxxpn001_call_macro_20220601-12

☑ A&R GRID

/opt/rstudio-connect/mnt/app/output/mkxxxxpn001_anrgrid_20220601-1253.xls

☑ WORKSPACE R OBJECT

/opt/rstudio-connect/mnt/app/workspace_user/mkxxxxpn001_robj_20220601-12

Figure 8. Document output specification

RESULTS

The completed prototype allows a user to enter study-specific information and customize parameters for seven pre-selected TLF mockups (tables: baseline characteristics, disposition, lab change, efficacy, adverse event, ae ich listing; figures: KM plot) using its GUI.

The template A&R grid is saved in the MS Excel XLS format. The template call program is saved as a .sas file. The Mock TLFs are saved in the RTF format and converted to PDF during display since most web browsers are not yet able to reliably display RTF content.

Upon the completion of its development, the prototype application was evaluated by statisticians. Feedback from the evaluations were then used to refine the application and identify future enhancements.

DISCUSSION

We have implemented a prototype of the automated mockup table and metadata generator that allows statisticians to specify parameters, generate and preview mockup table document, call program, which will greatly improve the user experience and also to ensure that the mockup shells are as expected. Global parameters, such as analysis populations and treatment groups, are specified centrally and hence can be kept consistent across all TLFs. The workspace containing parameter settings can be saved and then easily reloaded and adapted for other similar studies at a later time. This can also be used to build up a library of A&R metadata workspaces, which can reduce the chance of creating an A&R package from scratch, hence reduce the amount of turnaround time. To summarize, mockup table shells, template A&R grid and calling program with the corresponding mock TLFs are automatically generated by the application based on user specifications. This saves time and reduces human error from manual processing.

R Shiny provides a very reliable framework for generating user interface and developing applications such as this. Shiny has many active open-source development efforts, discussion forums and online example galleries, which are invaluable resources for finding example implementation for specific functionality and debugging during the application development.

Currently the application is developed in Shiny independent of the standard reporting macro development in SAS. Hence any update to a standard macro will require additional development effort for the application. Also, R Shiny programming is a new skill for some statistical programmers, which may require some upskilling effort. We expect the full-featured version of the application will be fairly complex, and detailed user documentation will be developed before it is deployed to production.

In the future version of the application, we hope to enhance its functionality by incorporating the following features:

- Enhance interface for browsing & selecting TLF templates
- Import settings from existing A&R grid and call program for TLF generation
- Create complete and more reliable call program code
- Offer flexibility in connecting the Shiny front-end with different back-ends implemented in various programming languages (e.g.: R & SAS)
- Use standard reporting programs as back-end to create mockups
- Build a library of A&R metadata for all TLFs from Global & TA standards libraries
- Provide ability to produce non-standard TLFs

CONCLUSION

Consolidating the A&R mock-up parameter specification and document creation processes via a single interactive application has the potential for streamlining the process and automating quality steps in creating the A&R deliverables. This prototype has demonstrated the benefits of the application. This will be further evaluated by broader set of users to get feedback before expanding the scope for production deployment.

REFERENCES

1. Gillespie, A., Kramlik, S., Sanjee, SR. (2020) “PROC Future Proof” Presented at PharmaSUG Virtual conference, San Francisco, CA. Retrieved from https://www.lexjansen.com/pharmasug/2020/SI/PharmaSUG-2020-SI-173.pdf

2. Busa, B., Murugesan P., Malcolm S. (2020) “Automation of TFL Generation using CDISC 360 Enriched Metadata” October 7th CDISC US Interchange. Retrieved from https://www.cdisc.org/sites/default/files/2021-08/Automation_of_TFL_Generation_using_CDISC_360_Enriched_Metadata.pdf
3. R Shiny: https://shiny.rstudio.com/
4. R: https://www.r-project.org/
5. Posit Connect Server: https://posit.co/products/enterprise/connect/

ACKNOWLEDGMENTS

The authors would like to thank Keaven Anderson, Ruchitbhai Patel, Sarad Nepal, Mukesh Patel, Uday Kiran Kandula, Yiwen Luo, Yilong Zhang and Endri Elnadav from Merck & Co., Inc., Rahway, NJ, USA, for their inputs and programming supports. We would also like to thank our IT colleagues Murali Talanayar, Loganathan Ramaswamy and Michael Wurst for helping with feasibility analysis of document driven programming. In addition, thanks to Merck management for their advice on this paper/presentation.

CONTACT INFORMATION

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

Jeff Cheng
Merck & Co., Inc., Rahway, NJ, USA
email: jeff.cheng@merck.com

Shunbing Zhao
Merck & Co., Inc., Rahway, NJ, USA
email: shunbing.zhao@merck.com

Guowei Wu
Merck & Co., Inc., Rahway, NJ, USA
email: guowei_wu@merck.com

Suhas R. Sanjee
Merck & Co., Inc., Rahway, NJ, USA
email: suhas_sanjee@merck.com