PharmaSUG 2021 - Paper AP-131

An R Shiny Application: Validated Analysis Results Using R and SAS

Abu Zafar Mohammad Sami,
mainanalytics GmbH, Germany

ABSTRACT

The production of a validated statistical manuscript in an accurate and timely way is a very challenging task when analyzing a clinical trial. Due to its sustainability and reliability, SAS has been the programming tool of choice over many years. However, R is gaining importance in this arena because of the flexible usage of packages.

The scope of this paper is to illustrate the process of an application tool that interactively produces the validated industry standard TLGs (Tables, Listings and Graphics) as well as data traceability for various statistical methods such as Regression and Survival Analysis, 2-way Tests, and HY's law. The user can interactively choose parameters to produce the desired outputs simultaneously through R and SAS scripts and thus cross check the results from the same input data. In this way, platform independent and accurate results are achievable in a quick manner. Moreover, the possibility of executing a SAS code from the R platform enhances data traceability.

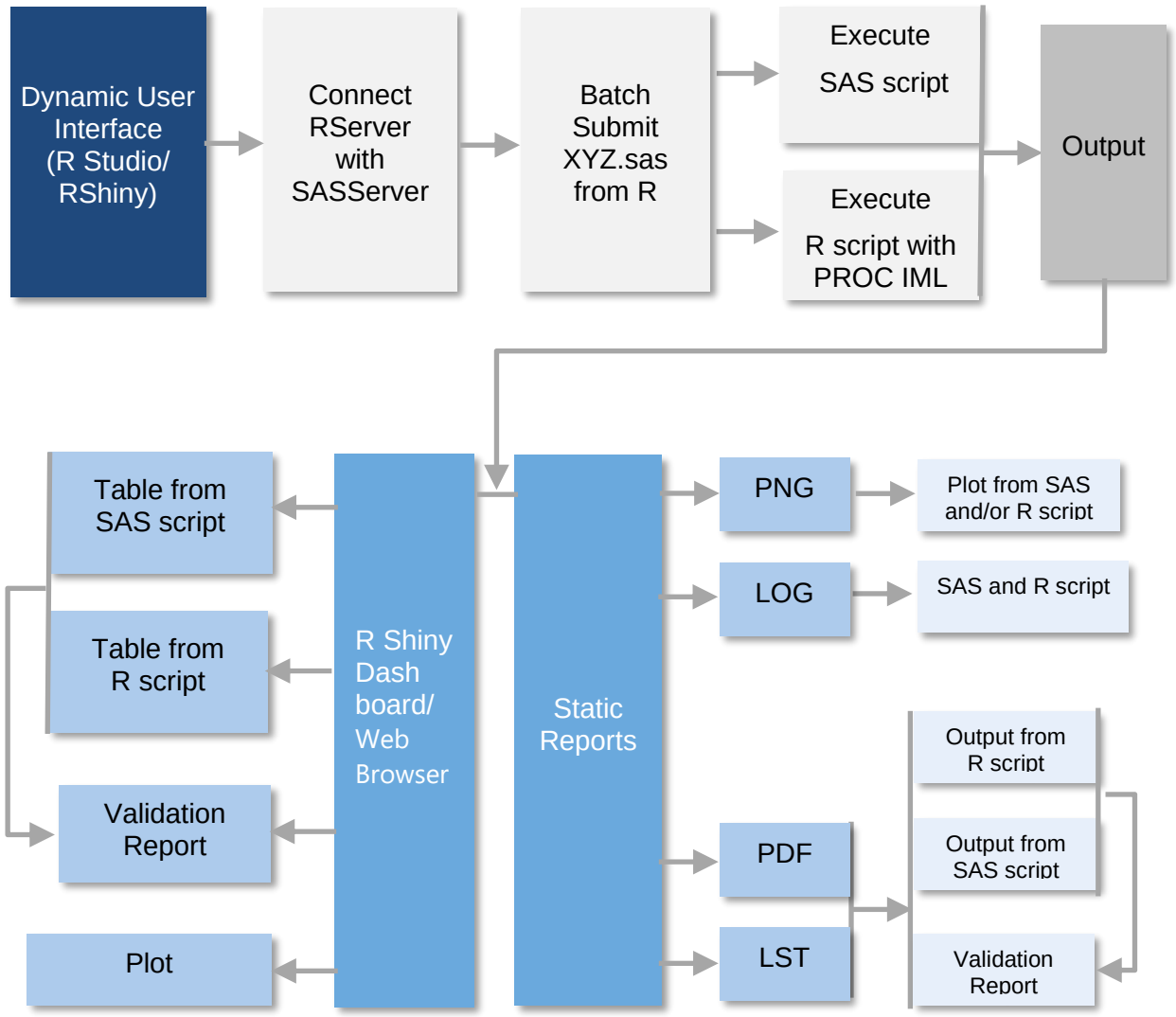
INTRODUCTION

R Shiny is an elegant and sophisticated tool to analyze clinical trial data interactively. On the one hand, the industry nowadays is in more and more practice of building Shiny apps because of its capability of loading data interactively, processing and analyzing the data with R scripts and packages, and producing immediate and/or static results. On the other hand, SAS is the most used programming language for the analysis of clinical trial data. The accuracy and reliability of SAS is trusted by the industry and regulatory agencies. Besides that, it ensures the data and process traceability through LOG files. Hence, the integration of both platforms could lead to obtain many attractive features.

This application tool uses the R Shiny framework as an interface. The variables and value level metadata of analysis datasets dynamically load from a specific project/study and the user can interactively select parameters of the underlying analysis from a drop-down list. As soon as the submit button is clicked, the back-end activity starts. The R script converts all the parameters of an underlying analysis to a temporary CSV file. At the same time, the RServer connects with the SASServer and executes a SAS script in batch submit mode. The SAS script, however, reads and converts the values of the CSV file to macro parameters and runs the SAS code as well as R code with PROC IML for the statistical method of the desired analysis.

The end user, however, receives the outputs in two formats. An immediate result of the underlying analysis is rendering in the Shiny Dashboard/Web browser consisting of the tables and, plots generated from both, the SAS and R system. Moreover, a comparison reporting of the generated numbers from both platforms will be displayed. Another type of outputs are static reports. To be a part of the clinical trial reporting, the static TLGs and STATDOCs are generated as LST/PNG and PDF files whereas LOG file and the validation report showing the comparison of the results ensure process traceability and accuracy. Thus, those platform independent validated results can be directly used for clinical trial reporting, e.g., for a manuscript, ad hoc analysis or for medical journals.

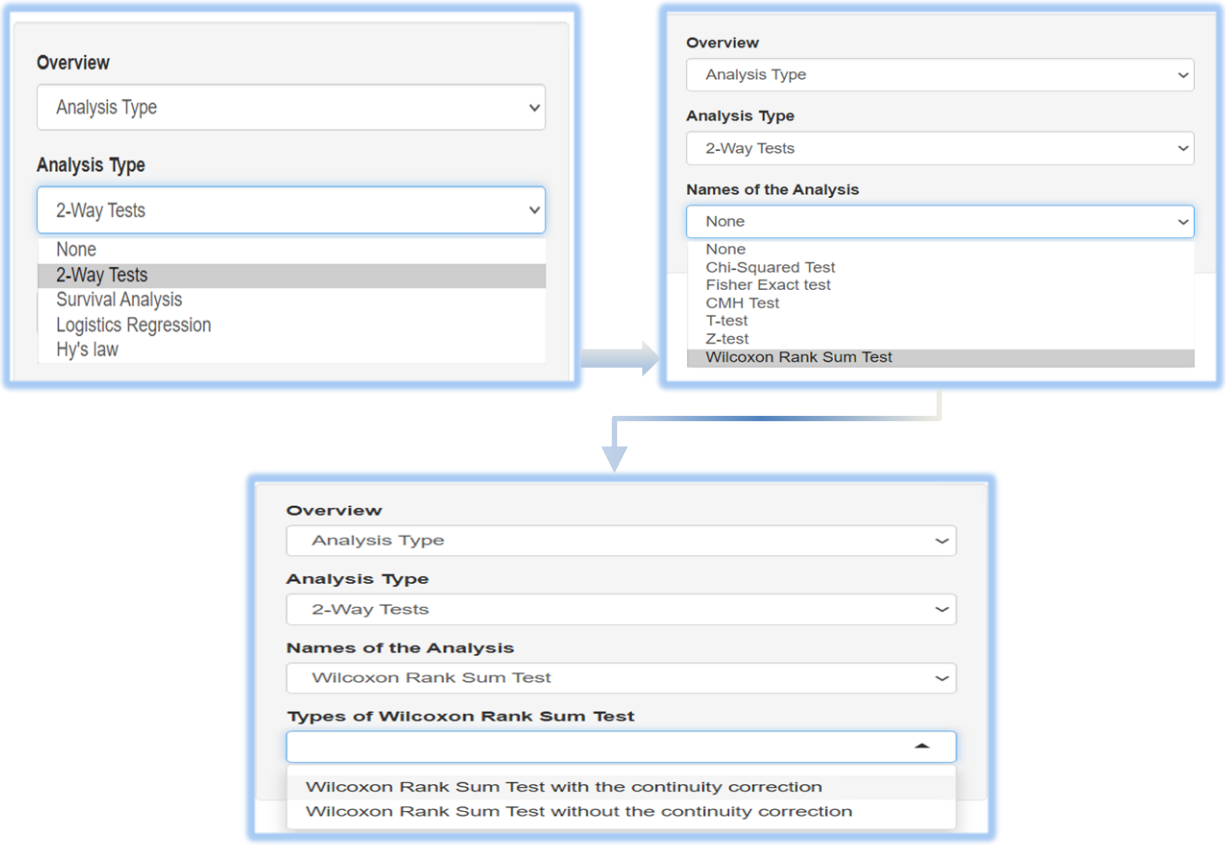
WORKFLOW OF THE APPLICATION TOOL



FRONT END: UI FRAMEWORK

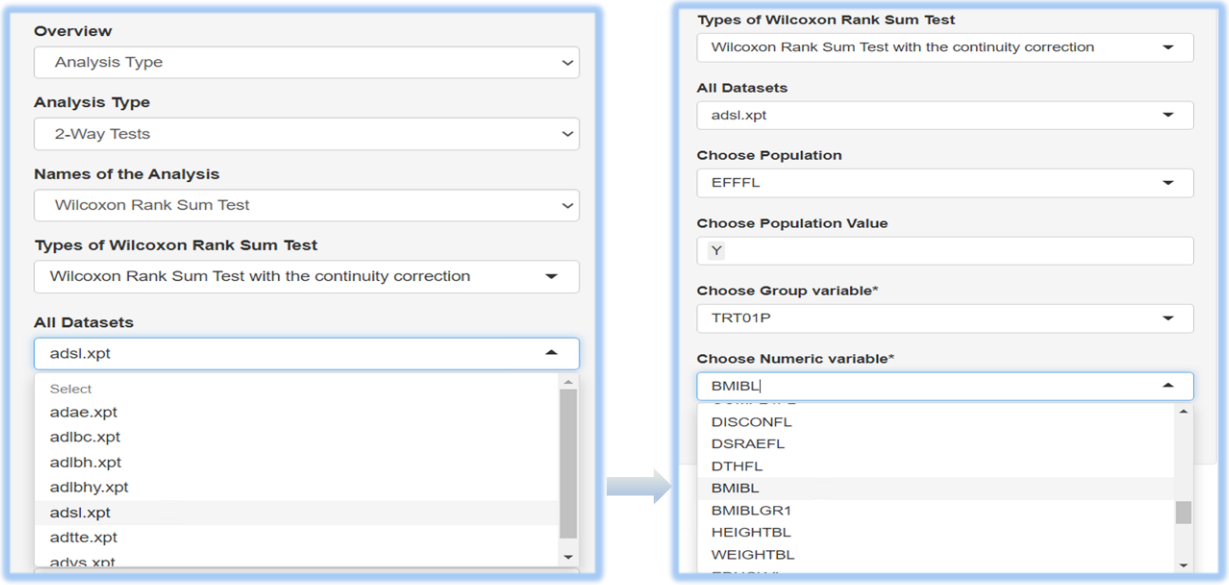
The user interface is developed under R Shiny/RStudio. In addition, R Shiny automatically generates HTML and CSS formatting which is compatible with any Web browser. Thus, an end user can use the apps through any web browser.
A prototype of using the app for an end user is demonstrated below:

Step 1: Selection of the Type of Statistical Analysis from Drop Down List



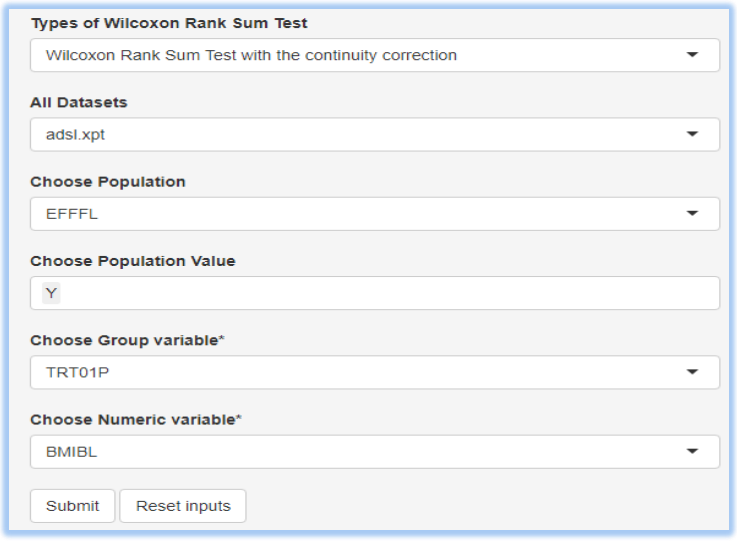
Step 2: Selection of the Parameters for the Underlaying Analysis (UA)

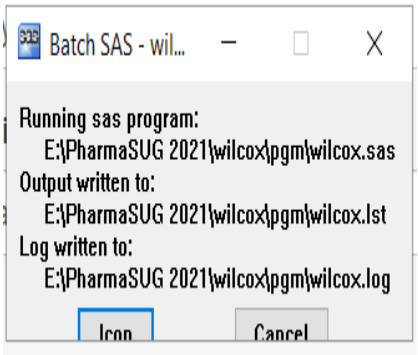
- Dynamic loading of the analysis datasets from study/project level.
- Dynamic loading of the variables and value level of metadata.



Step 3: Click Submit Button

- The execution of a SAS script embedded with SAS and R code of the selected statistical procedure at batch submit mode.
- The execution process also displays in the dashboard.





BACK END: THE ANALYSIS OF THE CLINCIAL TRIAL

Step 1: The Connectivity of RServer with SASServer

- A R script [1] runs to establish connectivity between both platforms.
- This script calls a SAS program of the underlying analysis (UA) from RStudio and executes in batch Submit Mode.

R script [1]:

```
callSAS <- function (f, path=getwd(), workDir=sprintf("E:",
                                        Sys.info()["user"])) {
  if (grepl(".sas$", f)) {
  #The location of SAS Executive file#
  exeFile <- "E:\\Apps\\SASHome\\SASFoundation\\9.4\\sas.exe"
  sasFile <- sprintf("%s\\%s", path, f)
  logFile <- sprintf("%s\\%s", path, gsub(".sas$", ".log", f))
  lstFile <- sprintf("%s\\%s", path, gsub(".sas$", ".lst", f))
  cmd <- sprintf("\"%s\" -sysin \"%s\" -log \"%s\" -print \"%s\"",
                 exeFile, sasFile, logFile, lstFile)
  system(cmd, invisible=FALSE)
  cat(readLines(sasFile), sep="\n")
```

Step 2: The Transmission of Parameters of the underlying analysis (UA) from R Shiny/ RStudio to SAS

- A R script [2] writes all the parameters that users have selected interactively from the R Shiny/Web browser UI into a CSV file.

R script [2]:

```
#The values from the selection boxes of R Shiny passing through
#input$[pointer of the boxes]
data<- input$dataset / group[1-2] <- input$group[1-2]
       population<- input$popu / population_value<- input$popval
a<- data.frame(data,group1,group2,population,population_value)
write.csv(as.matrix(a),"test.csv") #Writing into CSV file
```

- ***A CSV file is produced through R script [2]***

	A	B	C	D	E
1	DATA	GROUP1	GROUP2	POPULATION	POPULATION_VALUE
2	ADSL	TRT01P	BMIBL	EFFFL	Y

- A SAS script [1] reads all the required parameters of an underlying analysis (UA) from the CSV file and makes it compatible for using as macro parameters to execute the respective statistical methods in SAS procedures.

SAS script [1]:

```
#Read CSV File
 proc import datafile="E:/PharmaSUG 2021/wilcox/test.csv"
   out=readcsv dbms=csv replace; getnames=yes;

#Transpose Columns to Rows
proc transpose data=readcsv out=use_csv prefix=col_ name=param;
   var _all_;

#Track the parameters and value level of each parameter
proc sql;
   select distinct param, col_1
   into: _param separated by "#", : _value separated by "#"
   from use_csv;

#Count number of parameters
%let paramcount=%sysfunc(countw("&_param.","#"));

#Convert value of each parameter into a Macro Parameter
%macro params();
   %let i=1;
   %do i=1 %to &paramcount.;
     %let varnm=%scan("&_param.",&i,"#");
     %let varval=%scan("&_value.",&i,"#");
     data _null_; call symput("&varnm.","&varval."); run; %end;
%mend params;
```

Step 3: The execution of the Statistical Method of the underlying analysis (UA) in SAS and R

- A SAS procedure is executed for the desired statistical method through a SAS script [2]. The macro parameters derived from SAS script [1] are resolved inside the SAS procedure as: DATA=ADSL, POPULATION=EFFFL, POPULATION_VALUE=Y, GROUP1=TRTP01, GROUP2=BMIBL.

SAS script [2]:

```
proc npar1way WILCOXON correct=yes data=&data;
  where &population=”&population_value”
  class &group1;
  var   &group2;
  ods output WilcoxonTest=WORK.output_sas;
run;
```

- A R script is executed with the PROC IML procedure for the same desired statistical method through the SAS script [3]. Beforehand, the SAS configuration file (.CFG) needs to be updated with the following statements: -RLANG / -SET R_HOME "<your location of the R Software>".

SAS script [3]:

```
#Filter Dataset with condition
 data &data; set &data; where &population="&population_value";

#Start PROC IML Procedure to execute R Code
 proc iml;
  #Convert SAS Data to Dataframe
  run ExportDatasetToR("&data","mydata");

  submit / R;
    #Select required variables for the analysis
    mydata <- dplyr:: select(mydata,"&group1","&group2")
    colnames(mydata) <- c("group1","group2")

    #Call Analysis Procedure
    output$p.value <- wilcox.test(group2 ~ group1,data=mydata,
                      exact = FALSE, correct = TRUE)
    wilcox_pval<-as.data.frame(output$p.value)
 endsubmit;
 #Render Dataframe to SAS Dataset
 run ImportDataSetFromR("WORK.output_R","wilcox_pval");
 quit;
```

Step 4: The Comparison of the outcome results from both platforms

- A comparison SAS procedure is used to compare the results obtained from the SAS and R system.

```
proc compare base=WORK.output_SAS compare=WORK.output_R
   out=wilcox_diff OUTNOEQUAL;
run;
```

OUTPUT: THE REPORTING OF THE ANALYSIS RESULTS

The application tool produces two types of results. An instant result is rendering in the R Shiny dashboard. Another type of result is STATIC reporting which contains all the required files that could be a part of the clinical trial reporting, process validation, and data traceability.

INSTANT RESULT:

An end user can get an immediate result of the underlying analysis (UA) in the R Shiny dashboard that contains a validation report, analysis tables and graphics, generated from both SAS and R procedures. See below:

Table: Comparison of the results

	Compare Results between SAS and R Output
1	No unequal values were found. All values compared are exactly equal.

Showing 1 to 1 of 1 entries Previous 1 Next

Show 10 entries Search:

Table: Wilcoxon Rank Sum Test from R

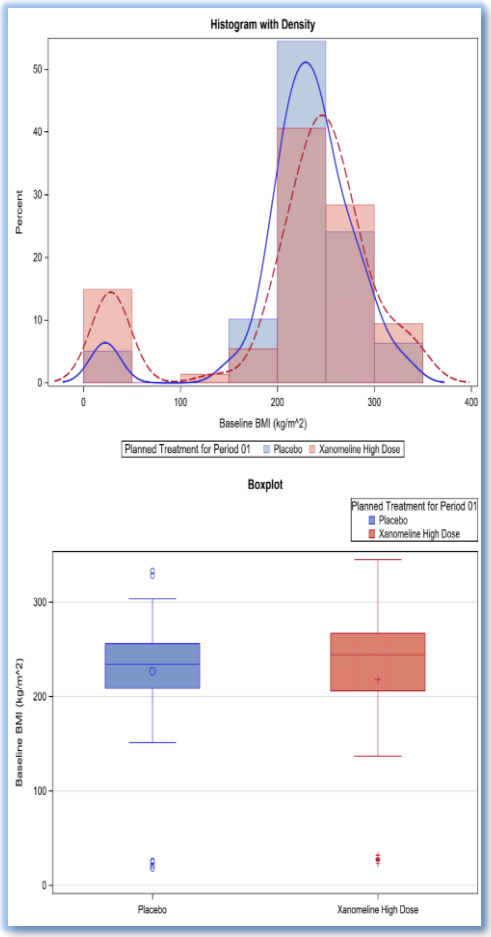
	Parameter: Baseline BMI (kg/m^2)	Placebo	Xanomeline High Dose	P-Value
1	Total subjects in Efficacy Population Flag	79 (51.63)	74 (48.37)	
2	Wilcoxon result			0.3939
3	N	79	74	
4	Mean	226.78	217.89	
5	Standard Deviation	59.35	87.93	
6	Standard Error	6.68	10.22	

Showing 1 to 6 of 6 entries Previous 1 Next

Show 10 entries Search:

Table: Wilcoxon Rank Sum Test from SAS

	Parameter: Baseline BMI (kg/m^2)	Placebo	Xanomeline High Dose	P-Value
1	Total subjects in Efficacy Population Flag	79 (51.63)	74 (48.37)	
2	Wilcoxon result			0.3939
3	N	79	74	
4	Mean	226.78	217.89	
5	Standard Deviation	59.35	87.93	
6	Standard Error	6.68	10.22	

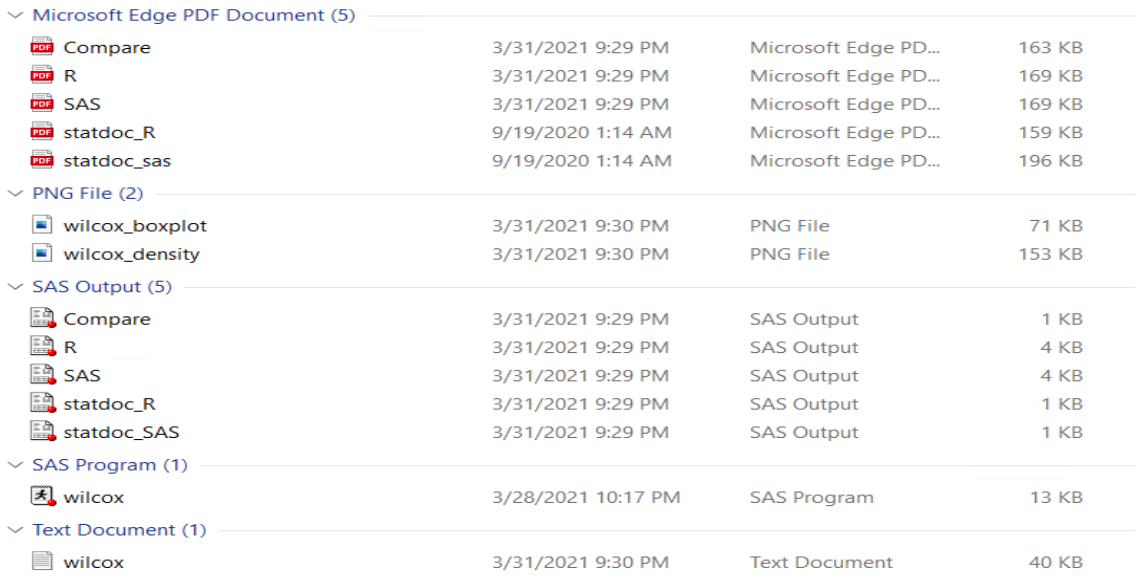


Hence, this gives users an instant summary and robustness in the analysis results.

STATIC RESULT:

Through this application tool, a branch of files for the underlying analysis (UA) is generated in the physical location. An appealing display of the analysis results with PROC REPORT, that could be potentially part of the clinical trial reporting is saved in LST/PNG/PDF format. On the other hand, a validation report is also generated in LST/PDF format. Finally, LOG file produces during batch submission ensure data traceability and process accuracy through checking for ERROR, WARNING, UNINITIALIZED etc. messages.

The set-up of the physical directory of static results are shown below:



Microsoft Edge PDF Document (5)			
Compare	3/31/2021 9:29 PM	Microsoft Edge PD...	163 KB
R	3/31/2021 9:29 PM	Microsoft Edge PD...	169 KB
SAS	3/31/2021 9:29 PM	Microsoft Edge PD...	169 KB
statdoc_R	9/19/2020 1:14 AM	Microsoft Edge PD...	159 KB
statdoc_sas	9/19/2020 1:14 AM	Microsoft Edge PD...	196 KB
PNG File (2)			
wilcox_boxplot	3/31/2021 9:30 PM	PNG File	71 KB
wilcox_density	3/31/2021 9:30 PM	PNG File	153 KB
SAS Output (5)			
Compare	3/31/2021 9:29 PM	SAS Output	1 KB
R	3/31/2021 9:29 PM	SAS Output	4 KB
SAS	3/31/2021 9:29 PM	SAS Output	4 KB
statdoc_R	3/31/2021 9:29 PM	SAS Output	1 KB
statdoc_SAS	3/31/2021 9:29 PM	SAS Output	1 KB
SAS Program (1)			
wilcox	3/28/2021 10:17 PM	SAS Program	13 KB
Text Document (1)			
wilcox	3/31/2021 9:30 PM	Text Document	40 KB

THE KEY BENEFITS

The user benefits from the following features of the application tool:

- Create analysis results interactively.
- Receive results from different platforms.
- Produce a validation report.
- Generate a log file to ensure data and process traceability.
- Use results directly for clinical trial reporting, e.g., for a manuscript, ad hoc analysis or for medical/scientific journals.
- Integrate any kind of analysis which can be done with SAS and R within this tool.

Thus, users can take advantage of the features of each of the programming languages. For example, use ggplot package of R for graphics or cross check the analysis results for a complex analysis from R packages/script with SAS PROC procedure.

CONCLUSION

An efficient production and appealing presentation of the outcome of a clinical trial analysis is in the interest of the industry. At the same time, the accuracy of the result and transparency of the process are uncompromisable. The whole idea behind the development of this tool is to deliver sustainable and reliable outputs in a quick manner. Nevertheless, a remarkable reduction of efforts and costs of a clinical trial analysis is achievable by using this application.

REFERENCES

1. Benjamin Chan (https://gist.github.com/benjamin-chan/73019d049763cff1d448)
2. Calling R Functions from SAS, PhUSE 2011.
   (https://www.lexjansen.com/phuse/2011/cs/CS07.pdf)
3. Using R in the SAS® System
   (https://www.lexjansen.com/wuss/2014/76_Final_Paper_PDF.pdf)

ACKNOWLEDGEMENT

I would like to thank Beate Hientzsch, Managing Director at mainanalytics GmbH and Katrin Besch, Marketing and Communication at mainanalytics GmbH for reviewing and providing valuable comments.

CONTACT INFORMATION

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

Abu Zafar Mohammad Sami

abu.sami@mainanalytics.de

www.mainanalytics.de