PharmaSUG 2019 - Paper AD048

Reimagining Statistical Reports with R Shiny

Sudharsan Dhanavel, Harinarayan Gopichandran, Cognizant Technology Solutions

ABSTRACT

Generating TLF outputs is one of the primary task for most of the statistical programmers across organizations. Generally, SAS programs and macros are involved to produce the outputs. Although there are multiple software packages or programming languages that could achieve this task. The focus on this paper is a solution based on R. This paper provides a brief background and overview of R Shiny package and its ability to solve the output generation and reviewing it for a statistical programmer.

A statistical programmer quite often receives multiple requests that results in several programming hours spent in altering the report parameters. Would it be cool if you point your output in dashboard and say, "Here you go it's an interactive Shiny app, you can alter the parameter or filter or subset as you wish to see and report right away appears in screen".

The Shiny package, by RStudio, let you specify input parameters in Graphical User Interface controls like sliders, drop-downs, and text fields; incorporating plots, tables, and summary outputs. The app logic written in R modifies the outputs immediately after the input in User Interface is changed.

INTRODUCTION

R is one of the widely used popular programming languages for Clinical data processing and analysis. It also has great flexibility and power to perform any kind of computation with the help of multiple packages released and contributed globally. If you are user of R, it is obvious you must have come across Shiny.

Shiny is a package from RStudio, which helps in building web applications thus, turning your analysis to interactive web platform for the other users. One could simply define an application created using Shiny as R + Interactive + Web. Although HTML, CSS and JavaScript libraries builds the web components, it is not necessary for one to have knowledge on those to get started creating shiny application. However understanding those will be an added advantage to quickly, built customized dashboard with minimum effort.

You can host your shiny app in www.shiny.io, which has both free and paid plans. Although the free server account comes with certain limitations. You can always go for Professional server that offers very well secured and scalable server environment handling multiple users.

WHAT DRIVES THE CHANGE

Although there are many related programming languages and tools for analysis and statistical programming this paper provides R based solution and its advantages as below:

- Shiny allows R developers generate interactive data visualizations that can be rendered on web and accessible from anywhere.
- The interactive and dynamic application is easy to use even for a non-programmer.
- Data scientist and Statistician no longer need support from the web experts to present their model build using R in web.
- In addition, with the ample of packages and global community support for R, development of Shiny applications becomes uncomplicated.
- The availability of charting libraries makes it an obvious choice for developing dashboards to represent the data in interactive way.

UNDERSTANDING THE FRAMEWORK

Any Shiny app rendered in a web page has a server running R behind.

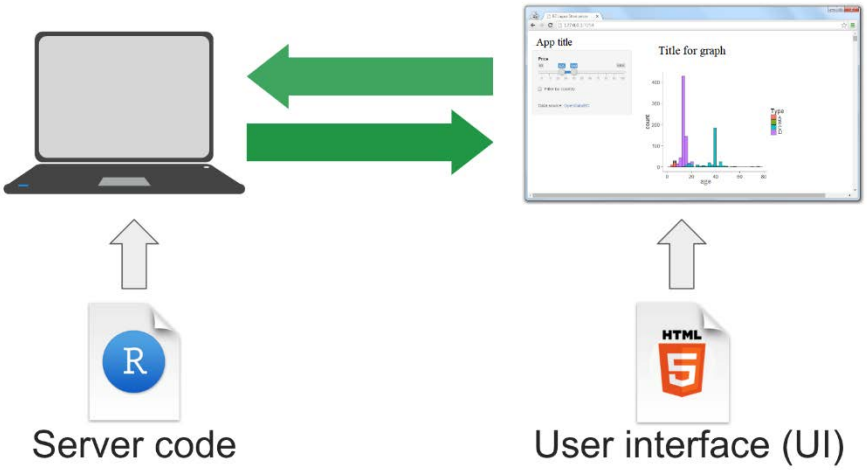

Figure-1

A Shiny application typically consist of two components:

1. ui.R: The function that builds the appearance and assemble the HTML user interface of the app.
2. server.R: The function with instructions on how to execute and rebuild the objects displayed in UI. It also contains the algorithms or models that works upon data.

An alternate way is to combine ui.R and server.R into a single file called app.R. This paper discuss the example of shiny app created using RStudio, which is an open-source integrated development environment for R.

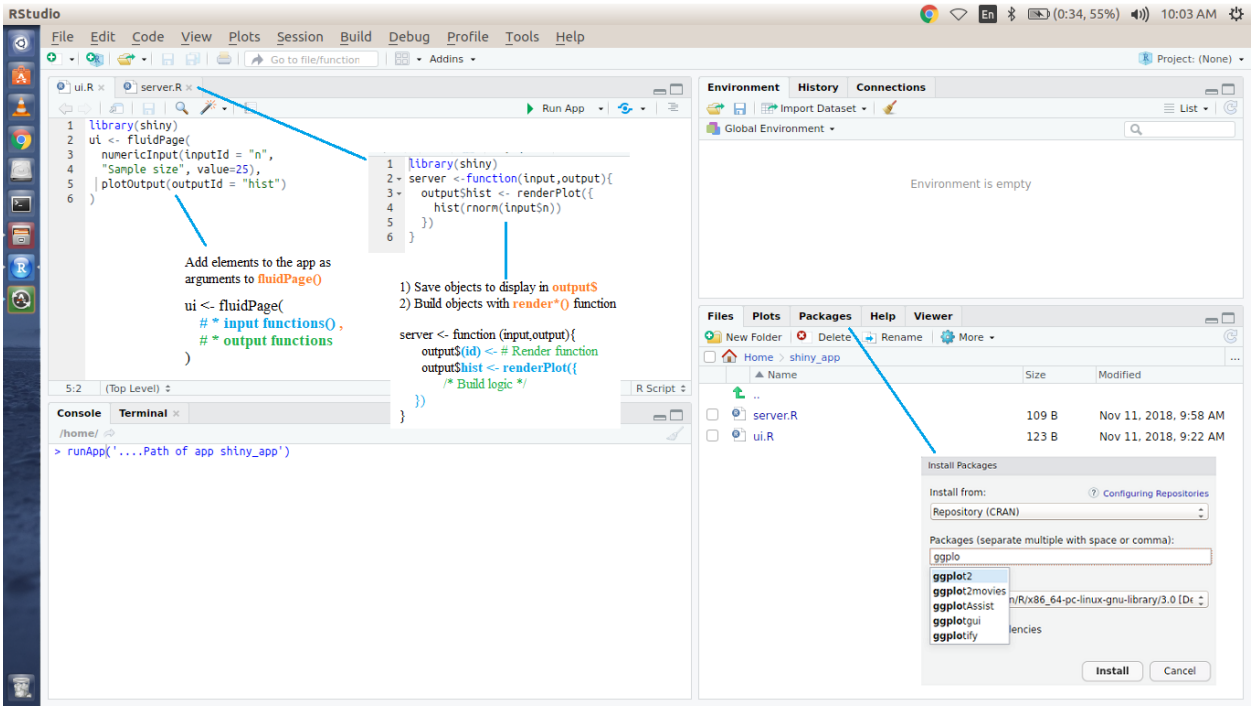

Figure-2

Inputs and outputs functions are used to build the shiny application, Inputs allows the user to provide values to the app and output objects are used to render the data, text, plots, and tables after the executing the logic when the input controls are modified. Shiny uses reactive programming model, so the framework is able to be fast, efficient, and robust. These input and output elements are added as parameters to the fluidPage() in ui.R.

There exist variety of input functions to create user interface elements that prompt the user for input values or interaction. Some of them are date input, select input, textbox, radio button, checkbox, file upload, slider etc. The output functions along with rendering function display the data in different types of output in the application. Some of them are plot, table, html and verbatim text rendered by renderPlot, renderTable, renderUI, and renderText functions respectively.

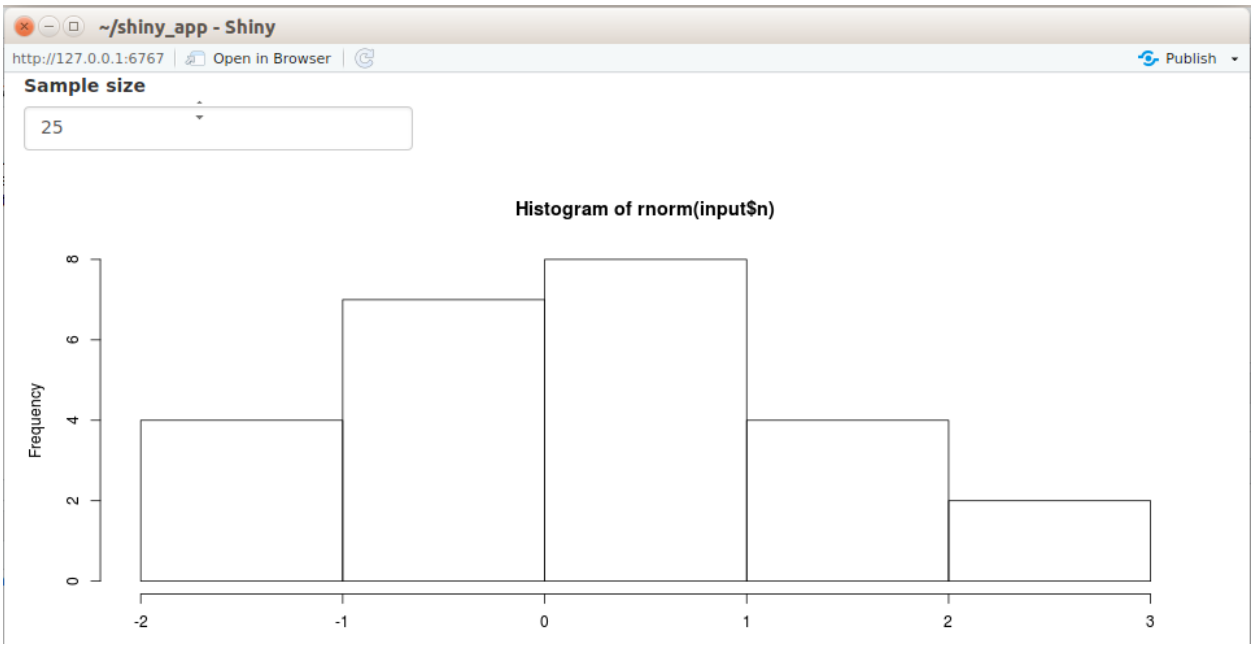

Figure-3

The code shown in figure-2 generates the output figure-3, a plot output rendered by renderPlot function providing the sample size in the numeric input. The R function rnorm generates a random value from the normal distribution. Modifying the sample size reactively, rebuilds the histogram.

Reactivity is what makes the Shiny apps responsive. It lets the app instantly update itself whenever the user makes a change. Shiny apps amazes users by running fast. Reactive expressions let you control which parts of your app update when, which prevents unnecessary computation that can slow down your app.

SHINY FOR A STATISTICAL PROGRAMMER

- Clinical trial reporting (Tables, Listings, Figures) using R has an absolute interest in recent past.
- As a result solid growth and its adoption, R in turn Shiny is becoming an in-demand skill for statistical programmers and biostatisticians.
- Interactive data exploration
- An easier access in terms of creation, usage and sharing of data/results.
- Relieve the burden of repetitive programming for set of deliverables.

SHINY APP: SUMMARY STATISTICS

This Shiny app will allows the user to produce summary statistic on selected parameters on uploaded analysis data. Summary statistics on demographic data is one of the routine output seen in every study deliverable.

On uploading the input data in (rdata, sas7bdat) the app renders the contents in datatable output. The population flag controls is used to filter the data, which in this case is safety flag; Select any other data from the library using dropdown control to merge with input data based on the required key variables. On each step, the processed data rebuilds the datatable output. The datatable comes with a search, filter and pagination options that helps to review data before generating summary.

This app contains four tabs:

1. **Data**: The resultant data rendered on the datatable control in grid format can be reviewed further using filter and search controls.
2. **Observation**: Displays the frequency count/ distinct values for each variables.

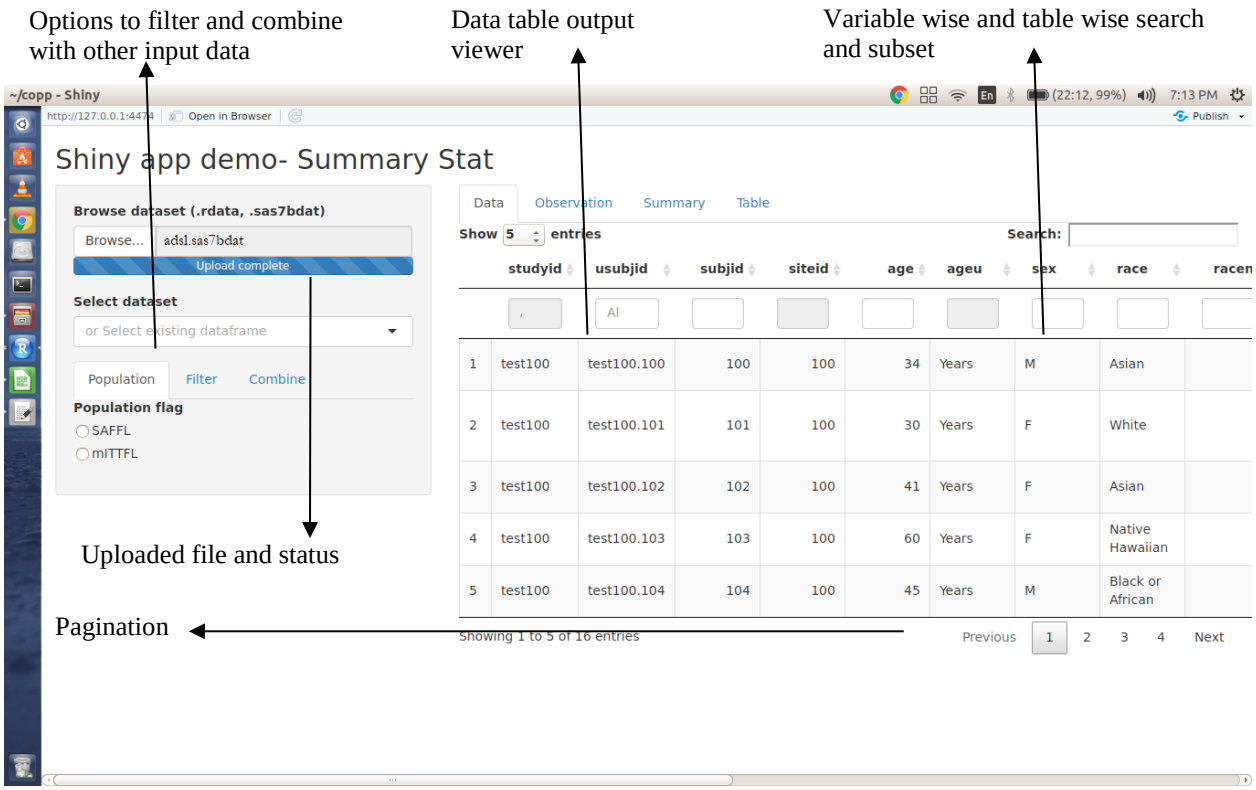

Figure-4

3. **Summary**: Overall summary (mean, median, min, max etc) on numeric variables and character variables. It also shows the structure of resultant data.
4. **Table**: Allows the user to select parameters (variable) to summarize and grouping variable to produce total. Although the app picks (trt01a) as default grouping variable, any variable replaces it when

changing the input.

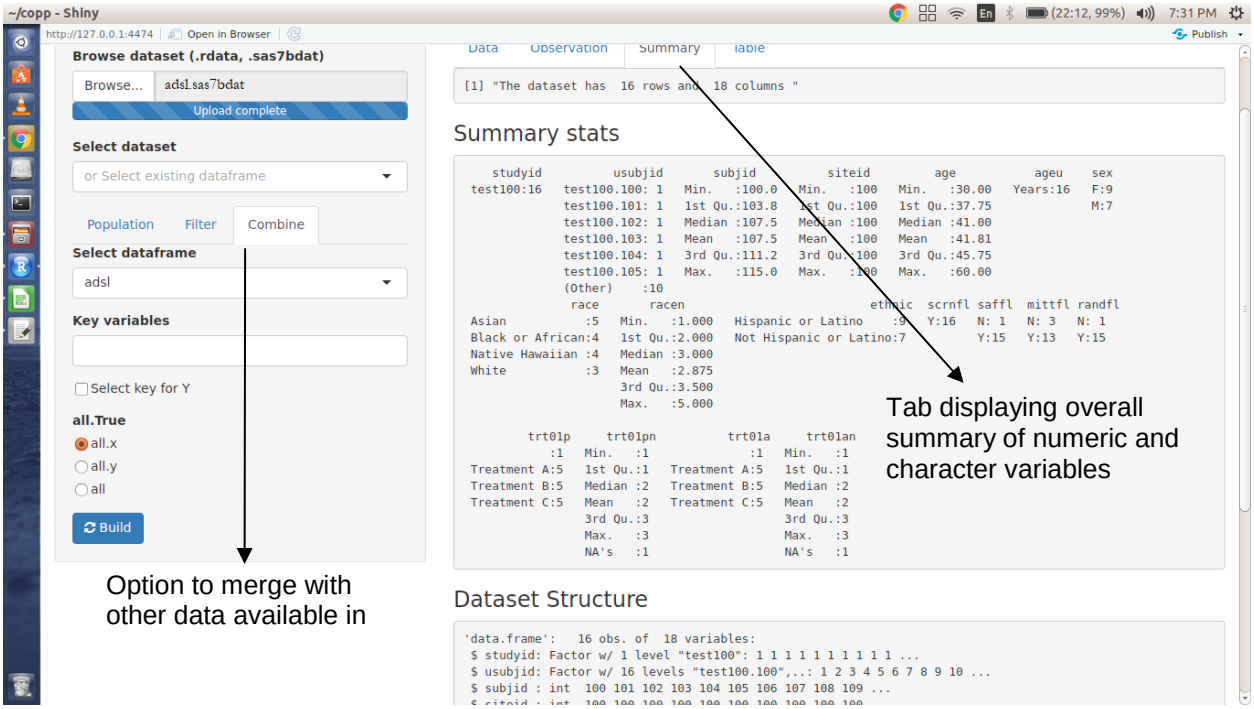

Figure-5

The dropdown control dynamically loads the variables from dataset processed in previous steps. The variables selected from the list will produce summary based on the datatype. ie) Character type only produces count n, whereas numeric variable produces Mean, Median, Q1,Q3, Min, Max and n

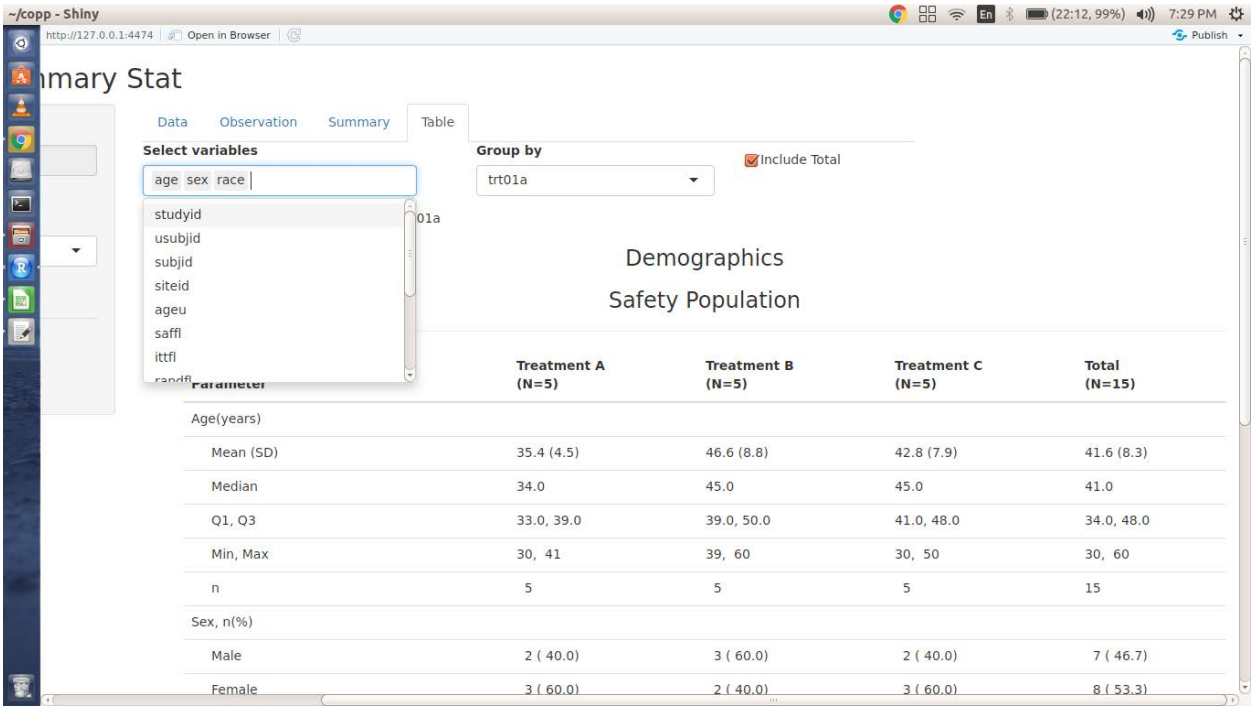

Figure-6

SHINY APP: FIGURE GENERATION

The ggplot2 and plotly packages can produce graphs with variety of graphical options made handy. The shiny app allows the user to upload the input data and plot the data on different type of graph. In addition, the app also supports merging, SQL execution, and deriving new variables (log, sqrt for demonstration). Although there are, plenty of graph options available this app has shown frequently used options for building the graph. The graph is downloadable to local computer.

https://youtu.be/uHdJ0sBHJ_w

CONCLUSION

Shiny apps provides lot more flexibility to produce the dynamic web apps that can produce any statistical output traditionally created using sas. The dynamic nature of Shiny makes it more efficient and build customized app automates routine task for deliverables.

REFERENCES

Paul Teetor. March 2011, R Cookbook. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472

Web Application Development with R using Shiny. Chris Beeley: Copyright © 2013 Packt Publishing
https://englianhu.files.wordpress.com/2016/01/web-application-development-with-r-using-shiny.pdf

Shiny - Tutorial https://shiny.rstudio.com/tutorial/

R powered web applications with Shiny http://zevross.com/blog/2016/04/19/r-powered-web-applications-with-shiny-a-tutorial-and-cheat-sheet-with-40-example-apps/

Interactive Web Apps with shiny Cheat Sheet https://shiny.rstudio.com/images/shiny-cheatsheet.pdf

https://shiny.rstudio.com/articles/understanding-reactivity.html

https://plot.ly/r/

REGULATORY COMPLIANCE REFERENCE FOR R AND SHINY

- *https://www.rstudio.com/wp-content/uploads/2014/06/RStudio-Shiny-Server-Pro-Validation.pdf*
- *https://www.r-project.org/doc/R-FDA.pdf*

CONTACT INFORMATION

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

Sudharsan Dhanavel
Cognizant Technology Solutions
Sudharsan.dhanavel@cognizant.com

Harinarayan Gopichandran
Cognizant Technology Solutions
Harinarayan.G@cognizant.com