PharmaSUG 2021 - Paper AD-164

R-Shiny based customized data viewer and search engine

Hrideep Antony, Syneos Health, Morrisville, North Carolina, USA

Aman Bahl, Syneos Health, Ontario, Canada

ABSTRACT

The number one reason why the Google search engine is most popular among all search engines is its ability to deliver millions of results in less than 0.19th of a second. In other words, users can view relevant information almost on a real-time basis. However, when it comes to the study data, the database viewers that are regularly used do not facilitate the real-time search that produces results interactively, as and when the users search for a keyword within the data.

This paper introduces an R-Shiny application that not only creates the results dynamically but also facilitates search at both database and variable level.

The application allows the user to select the variables that need to be displayed. It also gives the user control over the number of records to be displayed beside the ability to search the data as shown in figure 1 below. This search application supports multiple source formats such as .csv, .xls, .sas7bdat, etc.

INTRODUCTION

This R-Shiny based application uses function called “datatable()” which facilitates the search and filter options.The datatable function is from an R package named **DT** that provides an R interface to the JavaScript library and creates an HTML search widget that displays the data (a matrix or data frame) in the form of a viewer as shown in figure 1 below.

The viewer conducts a dynamic search across all the columns that are being selected (eg: 119) for the text that is entered inside the “search” tab. The search results are displayed as and when the text is typed without waiting for the user to hit “enter” or “submit”. This feature makes this viewer a very powerful utility especially because the users start seeing the search results at the very first keystroke without having to finish the text. Such a search mechanism is handy when the database involves free text eg: searching for an uncoded medication term or adverse event term.

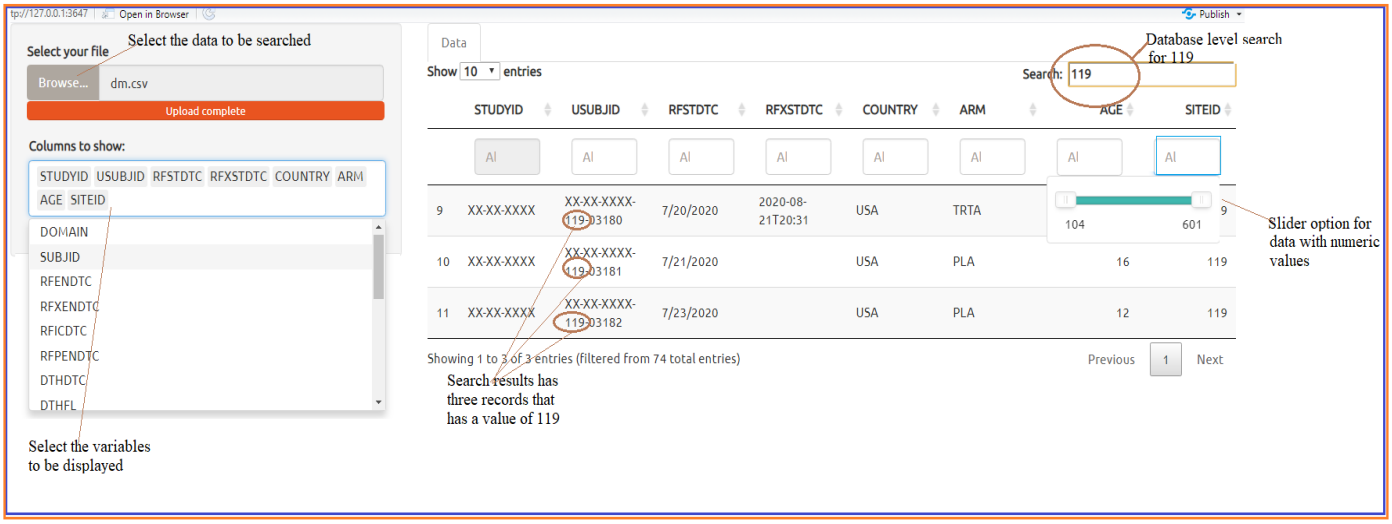

Figure 1. Project Success forecast

R-SHINY

Project Success forecast application is built using R-Shiny package.

Shiny is an R package that allows application development and comes as a built-in package available through CRAN that interacts well with RStudio. Shiny applications can be deployed as a webpage or a standalone desktop tool to view interactive data or perform statistical analysis.

A Shiny application typically consists of two components:

- UI.R: The function that builds the appearance and assembles the HTML user interface of the app.
- Server Code: The function with instructions on how to execute and rebuild the objects displayed in UI. It also contains the algorithms or models that build upon data

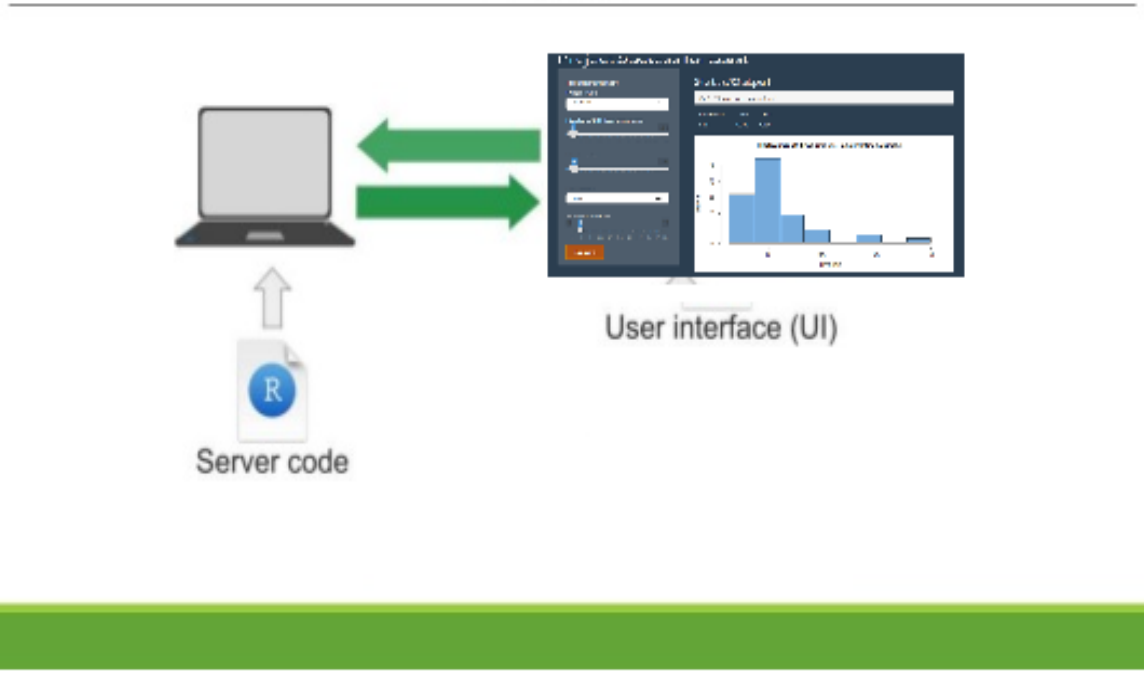

Figure 2. Shiny components

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

There exist a 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 the 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.

Further information about building reactive web apps using R-shiny can be found in the shiny [cheet sheet url.]

DT PACKAGE AND DATATABLE LIBRARY:

DT is an interface to the JavaScript library DataTables based on the html widgets framework, to present rectangular R data objects (such as data frames and matrices) as HTML tables. Using the datatable function we can filter, search, and sort the data in the table.

DataTables comes with a global search box by default to which column filters can be added using the filter argument. For numeric/date/time columns, range sliders are used to filter rows within ranges which is very useful for filtering numeric results.

The overall syntax of the "datatable" function that is used in this application can be found in figure 3 below. Please refer to the R-documentation link for complete details.

```
datatable(
  data,
  options = list(),
  class = "display",
  callback = JS("return table;"),
  rownames,
  colnames,
  container,
  caption = NULL,
  filter = c("none", "bottom", "top"),
  escape = TRUE,
  style = "default",
  width = NULL,
  height = NULL,
  elementId = NULL,
  fillContainer = getOption("DT.fillContainer", NULL),
  autoHideNavigation = getOption("DT.autoHideNavigation", NULL),
  selection = c("multiple", "single", "none"),
  extensions = list(),
  plugins = NULL,
  editable = FALSE
)
```

Figure 3: datatable syntax

RSHINY CODE

R shiny code to create the data viewer is provided below.

```
library(shiny)
library(ggplot2)  # for the dataset
library("safetyGraphics")
library(shinythemes)

####################################
# User interface                   #
####################################
ui <- fluidPage( theme = shinytheme("united"),
  title = "Examples of DataTables" ,
  sidebarLayout(
    sidebarPanel(
      fileInput('file1', 'Select your file',
                accept = c(
                  'text/csv',
                  'text/comma-separated-values','.csv',
                  '.sas7bdat'
                )
      ),

      selectInput("varlist","Columns to show:",choices=c(colnames(data())),
       multiple = TRUE)
    ),
    mainPanel(
      tabsetPanel(
        id = 'dataset',
        tabPanel("Data", DT::dataTableOutput("mytable1"))
      )
    )
  )
)
```

```
###################################
# Server                          #
###################################

server <- function(input, output, session) {

  # read in data
  data1 <- reactive({
    if(is.null(input$file1)) return(NULL)
    inFile <- input$file1
    read.csv(inFile$datapath)
  })

  # choose columns to display
  observeEvent(input$file1,{
    updateSelectInput(session,"varlist",choices=c(colnames(data1())))
  })

  new_data<-reactive({
    raw_data <- data1()
    DT::datatable(raw_data[, input$varlist, drop = FALSE],filter = "top" , selection = 'multiple' ,escape=FALSE )
  })

  output$mytable1 <- DT::renderDataTable({
    new_data()
  })
}
###################################
# Create the shiny app            #
###################################
shinyApp(ui, server)
```

CONCLUSION

The web application framework provided by Rshiny acts as a user interface medium to navigate and select the data to be explored. This Rshiny interface along with the widgets framework provided by the DT package makes this application an excellent search and dataset viewer with very dynamic filtering capabilities. This application is being further expanded to create dynamic dashboards using markdown language that will provide a visual delight for the users using 3D summary plots and descriptive statistics of the data.

REFERENCES

Datatable: Create an HTML table widget using the DataTables library

https://www.rdocumentation.org/packages/DT/versions/0.17/topics/datatable

Package 'DT'

https://cran.r-project.org/web/packages/DT/DT.pdf

ACKNOWLEDGMENTS

Sincere thanks to Steve Benjamin, Director Statistical Programming, Biostatistics and Global Contracts and Aman Bahl, Associate Director, Statistical Programming, Clinical Division for their vision, great leadership, persistent support, and encouragement throughout and for their valuable assistance in reviewing this paper.

CONTACT INFORMATION

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

Hrideep Antony
Principal Statistical Programmer, Clinical Division, Syneos Health
Work Phone: +1- 984 459 4785
Email: hrideep.antony@syneoshealth.com
Web: http://www.syneoshealth.com

Aman Bahl
Associate Director, Statistical Programming, Clinical Division, Syneos Health
Phone: +1-905 399 6715
E-mail: Aman.Bahl@syneoshealth.com
Web: http://www.syneoshealth.com