

PharmaSUG China 2016-74

Big Insights in Small Data with RStudio Shiny

Mina Chen, Roche Product Development in Asia Pacific, Shanghai, China

ABSTRACT

Accelerating analysis and faster data interpretation helps to stay competitive in the drug development field. It requires statistical programmers to explore and adapt new, different and suitable approaches to delivery of data analysis according to the stakeholder's requirement. A key approach to data analytics is our increasing use of visualization tools in conjunction with analyst skills to support faster decision-making.

As customer demands evolving, statistical programmers are required to proactively communicate and interact with key stakeholders and clearly identify their needs. Our practice is to utilize and explore a newly developed data analysis approach 'RStudio Shiny' to support ad-hoc and exploratory analysis based on specific requirement. This paper outlines how statistical programmers support effective data review and fast decision-making by using RStudio Shiny application.

INTRODUCTION

There're great needs from stakeholders as clinical trials are getting more complex. In this case, new roles are emerging such as data analysts for statistical programmers that indicate new skill needs, building on the current work and related technical skills. In this paper we will discuss some of the applications we have developed using RStudio Shiny to analyze and visualize clinical trial data. The "Shiny" package which is designed by RStudio, allows access to the diverse and powerful statistical methods implemented in R for users without programming experience or even statistical knowledge.

The "Shiny" package enables statistical programmers explore and explain data via interactive webpage GUIs, which reduces the interaction with complex statistical tools. The ease of developing Shiny applications allows programmers to easily design very specific and fancy GUI layouts that can ensure correct application of the statistical methods.

In clinical trials we have make use of Shiny applications for exploratory data review and ad-hoc analysis. In the following sections, we will give a brief introduction of RStudio Shiny including what is Shiny, how to use Shiny, as well as the application development flow. Then we will present an example to demonstrate how we use Shiny application to support fast decision making. The example is an application that supports biostatistician to calculate data-driven conditional probability via Shiny app.

COUPLE OF WORDS ABOUT RSTUDIO SHINY

"Shiny" is a new package from RStudio that makes it incredibly easy to build interactive web applications with R. It provides an easy way to make an interactive web page, and that web page can seamlessly interact with R and display R objects (plots, tables). Although it may sound scary because of the words "web pages", it's geared to programmers who have zero experience with web development, and programmers do not need to know any HTML/CSS/JavaScript. To get a sense of the wide range of things you can do with Shiny, you can visit <http://shiny.rstudio.com/>, which provides tutorials, articles and references of how to build Shiny application. Before reading any of the examples below, you'll need to know a few things about R and "Shiny" package. This is not a paper on how to program in R, but there are some basic knowledge needed in order to understand what is going on.

The main features of RStudio Shiny include:

- § Build useful web applications with only a few lines of code—no JavaScript required.
- § Shiny applications are automatically “live” in the same way that spreadsheets are live. Outputs change instantly as users modify inputs, without requiring a reload of the browser.
- § Shiny user interfaces can be built entirely using R, or can be written directly in HTML, CSS, and JavaScript for more flexibility.
- § Works in any R environment (Rgui for Windows, RStudio, etc.).
- § A highly customizable slider widget with built-in support for animation.
- § Pre-built output widgets for displaying plots, tables, and printed output of R objects.
- § Develop and redistribute your own Shiny widgets that other developers can easily drop into their own applications.

SETUP AND DESIGN

In order to build the application you should make sure you have RStudio software and the “Shiny” R package installed in your computer. Development and testing of R program was done in RStudio development environment. Other R packages like “dplyr”, “ggplot2”, “gridExtra” and “reshape2” should also be installed to aid the application development.

SHINY APPLICATION WORKFLOW

The logic of RStudio Shiny relies on the two notions that are very close to the web ones: client side and server side. Client side is the component in interaction with the end-users. Server side is the component through what your R program receives requests from users, runs R code and then send the response to the client side.

Based on this principle, Shiny requires two R scripts: one for the User Interface, named ui.r, and one for the Server side, named server.r. What you want to display on the interface needs to be declared in the ui.r script; how this thing is done needs to be defined in server.r. Below is a typical workflow of a Shiny application:

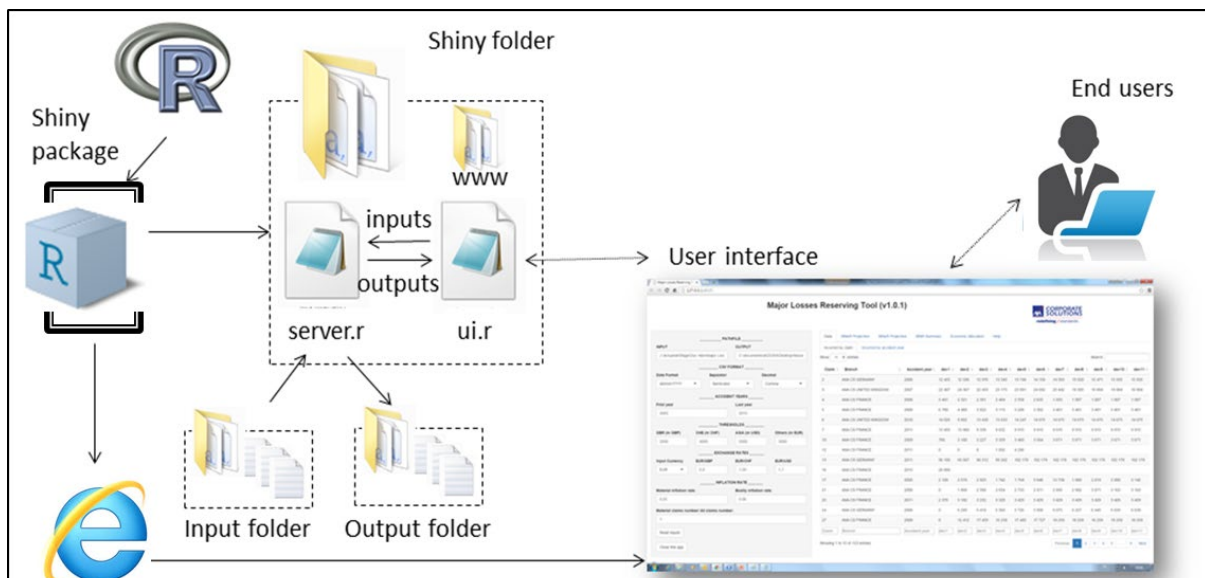


Figure 1 Shiny application workflow

R calls the Shiny app in a main script, with the launching of Shiny package. Normally a Shiny folder was created which contains two principal elements: ui.r and server.r, as showed in figure 1. Sometimes a third element named 'www' also be included, which is a sub-folder with graphical elements (images, icons, etc.) and file (pdf) that enables programmer to display on the application. One thing need to notice is that these three elements need to be named exactly that way. In addition, the server,r can also call other R scripts.

The user interface can be displayed on the default browser of RStudio if you run Shiny through it, or on a web browser (if you are using Internet explorer version < 10, you could get into trouble with Shiny). This user interface works exactly like a web page. There are buttons, input text, etc. for you to make your decision and there will be output (graphs, images, tables, etc.) displayed on the main panel, or separately on different tabs/panels.

APPLICATION SHOWCASE

In this section we will introduce an application we developed and provide the screenshots accordingly.

EXAMPLE APPLICATION: DATA DRIVEN CONDITIONAL PROBABILITY CALCULATION

Generally, Biostatisticians perform Data Driven Conditional Probability (DDCP) to inform the project on the Probability of Technical Success (PTS). However, there are many scenarios, which was very due to different study endpoints or different data distribution. It's time consuming for Biostats to gain DDCP under mixed conditions and present the results concisely and clearly to the study team.

As a statistical programmer, we are upskilling new data visualization technical and aim to support biostatistician to calculate data-driven conditional probability (DDCP) using RStudio Shiny.

Here is the work flow of how we identify the stakeholder's need:

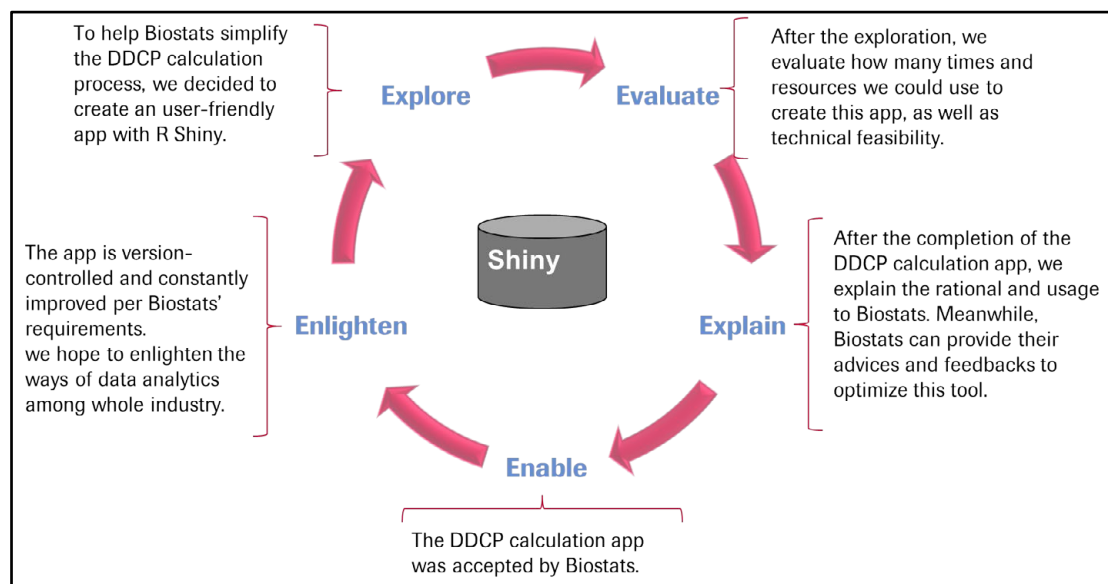


Figure 2 "DDCP" application workflow

Figure 2 shows the general working flow of the DDCP application development, which contains 5 stages: explore, evaluate, explain, enable and enlighten. At the very beginning, we had back and forth discussions with biostatistician and make sure that we can identify their real needs. Then after the app was developed, we did some preliminary test and collected feedback from end users. It's an ongoing process and there may be more features added as customer demands evolving.

Figure 3 is the main interface of the DDCP calculation app which includes 4 tabs: instruction tab and 3 scenario tabs. The application is developed for 3 scenarios: time to event endpoint, binary endpoint and normal distribution. Different formulas are specified for different scenarios.

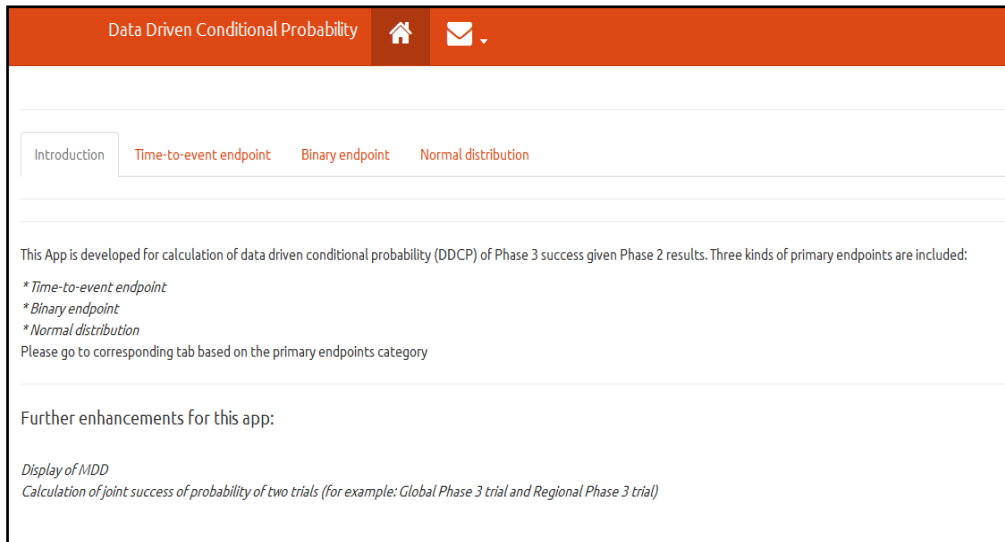


Figure 3 the "Introduction" tab

Figure 4 shows the results of one of the scenarios. Take the "Time-to-event endpoint" as an example. The control panel on the left of the app includes conditions you should enter before calculation. The outputs contain tables, plots, as well as summary notes. The plot is interactive and the values will show up if you put your mouse on the specific points in the plot.

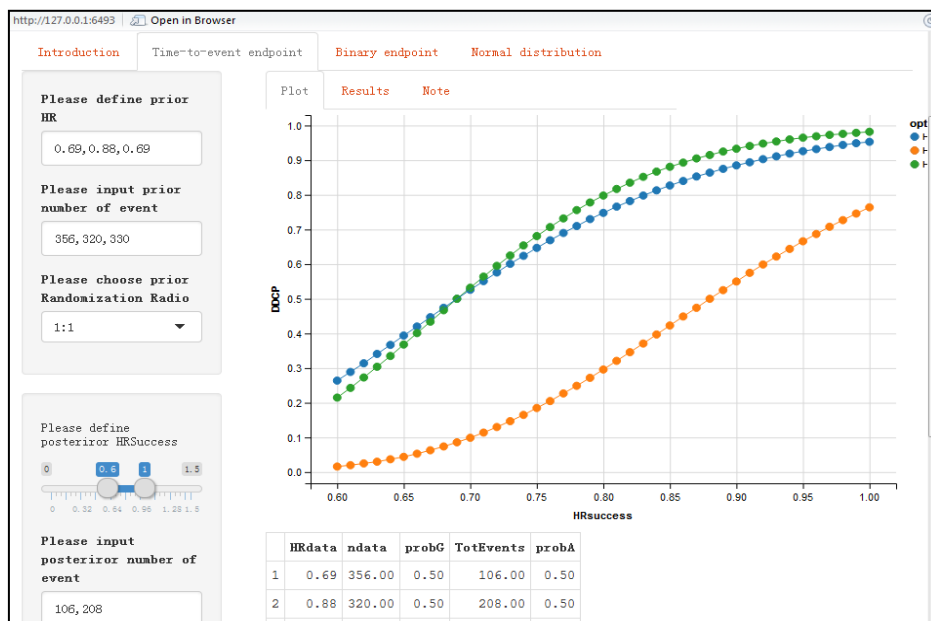


Figure 4 the "Time-to-event endpoint" tab

Once you've written your own app, you can distribute it for others to run on their own computers. They can download and run Shiny apps with a single R command. This requires that they have R and Shiny installed on

their computers. It's also possible to share Shiny app with people who do not have R (and have no intention of getting it) via a web page. If you host the app at its own URL, users can visit the app and no need to worry about the R code.

CONCLUSION

Shiny enables programmers to write powerful interactive web applications entirely in R. Using RStudio you create a user interface and Shiny compiles your code into the HTML, CSS and JavaScript needed to display your application on the web. What makes a Shiny app particularly powerful is that it can execute R code on the backend so your app can perform any R calculation you can run on your PC. Perhaps you want your app to manipulate a dataset based on user inputs. Or maybe you want your app to run survival models or complicated statistical methods based on user-selected data. In either case, Shiny app can help.

Creating and running simple web application is relatively easy and there are great resources for doing this. The examples and discussions offered in this paper demonstrate how to support stakeholders better understand and explore data by using Shiny application. As customer demands evolving, statistical programmers can support fast decision making in a more efficient way.

REFERENCES

A guide to getting started in R for SAS programmers, available at: <http://r4stats.com>.

Shiny by RStudio tutorial, available at: <http://shiny.rstudio.com/>.

Tutorial: Building 'Shiny' Applications with R, available at: <http://rstudio.github.io/shiny/tutorial/>.

The R Project for Statistical Computing, available at: <http://www.r-project.org>.

ACKNOWLEDGMENT

Thanks to my colleagues Xiuting Mi, Lynda Li and Hangjun Dai for their help in building the DDCP application together.

CONTACT INFORMATION

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

Name: Mina Chen

Enterprise: Roche (China) Holding Ltd.

Address: 11 Building, floor 4, 1100 Long Dong Ave. Shanghai

City, State ZIP: Shanghai, China

E-mail: mina.chen@roche.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.