Paper DV07

Using R/Shiny to Create an Interactive-Visualization Patient Profile

Elva Chen, Corvus Pharmaceuticals, Burlingame, USA
Brian Munneke, Corvus Pharmaceuticals, Burlingame, USA

ABSTRACT

The ability to visualize data dynamically is vital for making the best and most accurate decisions. Shiny is an open source R package that combines the computational power of R and modern visualization techniques to create interactive applications. We have applied various functionalities of R/Shiny to create a graphical, interactive patient profile. The aim of this paper is to demonstrate how R/Shiny can be a great tool for building reader-friendly patient profiles, and improve the way we access, review and interact with data.

INTRODUCTION

Patient profiles are a collection of information that allows sponsors and investigators the ability to keep track of subject information collected via an eCRF system and maintained in a clinical database. Each patient profile is essentially a complete data extract that includes variables such as the subject's demographics, vital signs, medical history, treatments, lab results, concomitant medicines, adverse events and other domains included in clinical database. Most statistical programmers use SAS to create many listings along with some limited graphics to serve as patient profiles when preparing data for review. The long list of static outputs is often cumbersome and confusing for reviewers. We will present an interactive visualization of a patient profile that dynamically displays graphs and tabular outputs of the data as selected by the reviewer. The interactivity of the patient profiles provides a quick and easy way to gain an understanding of the current status of any subject or set of clinical variables in the clinical database.

A BRIEF OVERVIEW OF R/SHINY

Shiny is a framework to develop applications comes as a built-in package available through CRAN that interacts nicely with the RStudio IDE. Shiny applications can be deployed as a webpage or, a standalone desktop tool to interactively view data or perform statistical analysis. As additional examples, other groups have successfully used the R/ Shiny framework to create very useful interactive tools such as the SafetyGraphics Explorer created by the Interactive Safety Graphics (ISG) subteam of the ASA Biopharm-DIA Safety Working Group.

A basic Shiny application has two parts: a user interface (UI) and a server function. Shiny library needs to be called in the beginning of the program that combines the UI and server code blocks together to form a working application:

```
library(shiny)
ui <- fluidPage()
server <- function (input, output){}
shinyApp(ui=ui, server=server)
```

In this paper, we use the RStudio console to read in SAS data sets in CDISC standard domain model then create graphs and display data with Shiny. We will use the DM, AE, LB, and VS data sets and will focus on functionalities of R/Shiny and their interactive components

DYNAMIC DATA SELECTION

Shiny has several designs of user-interface for UI layout. We choose *navbarPage* to organize domains of data. Fig. 1 displays the overall look of *navbarPage* dashboard of our application titled 'Patient Profile using R/Shiny". We use Demographics, Vital Signs, Adverse Events and Laboratory as examples. The *navbarPage* can extend to include more domains in one dashboard. User selects what data domain to display by clicking on manual bar such as 'Demographics'.

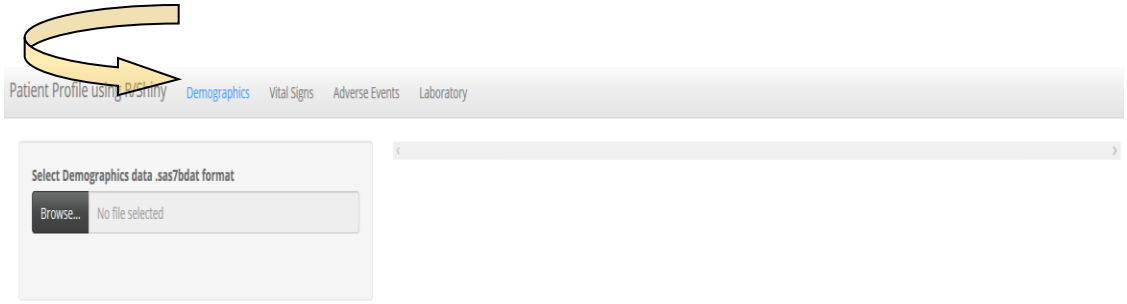


Fig.1

Clicking on 'Browse' will let user to scroll through folders to select dataset. Data will appear simultaneously after dataset selected. Fig. 2 shows the demographics data after 'dm2' data is selected.

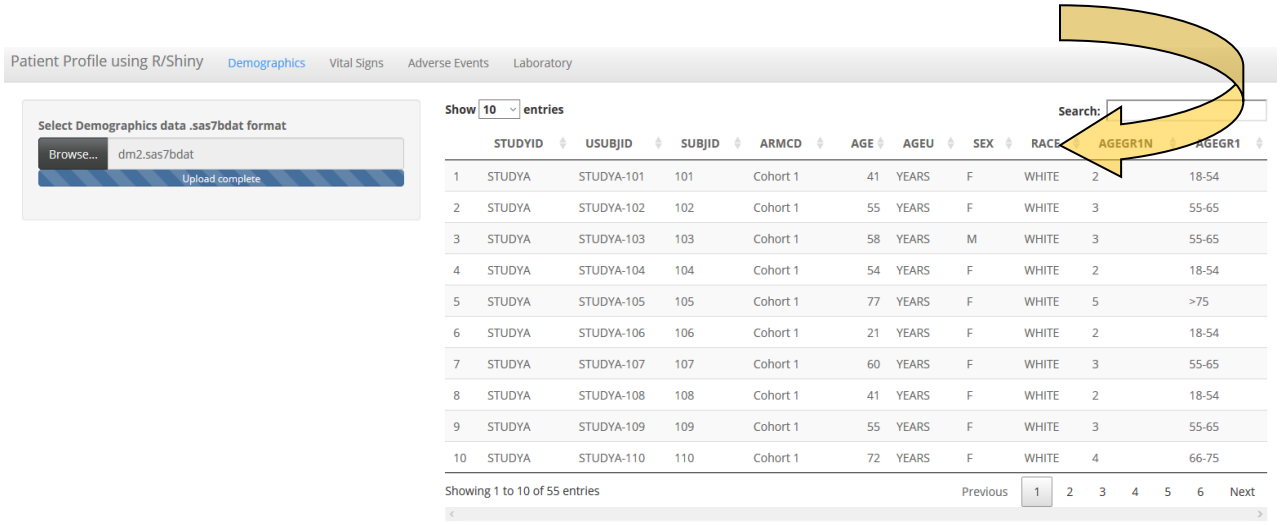


Fig.2

DYNAMIC SUBJECT SELECTION AND GRAPHICAL DISPALY

Each domain in clinical data contains domain-specific information for all subjects. In Fig. 3, we use the Vital Signs domain as an example to introduce interactive functionality that allows the user to choose a subject from the selected dataset. As user selects a subject, the R/Shiny program, executing quietly in the background will

simultaneously display the scatterplot. In the example, we use the ggplot2 package to plot Systolic and Diastolic Blood Pressure over study day. We can also place Systolic and Diastolic plots together in a single panel graphical object. As the user updates the subject selection, the plots are updated dynamically.

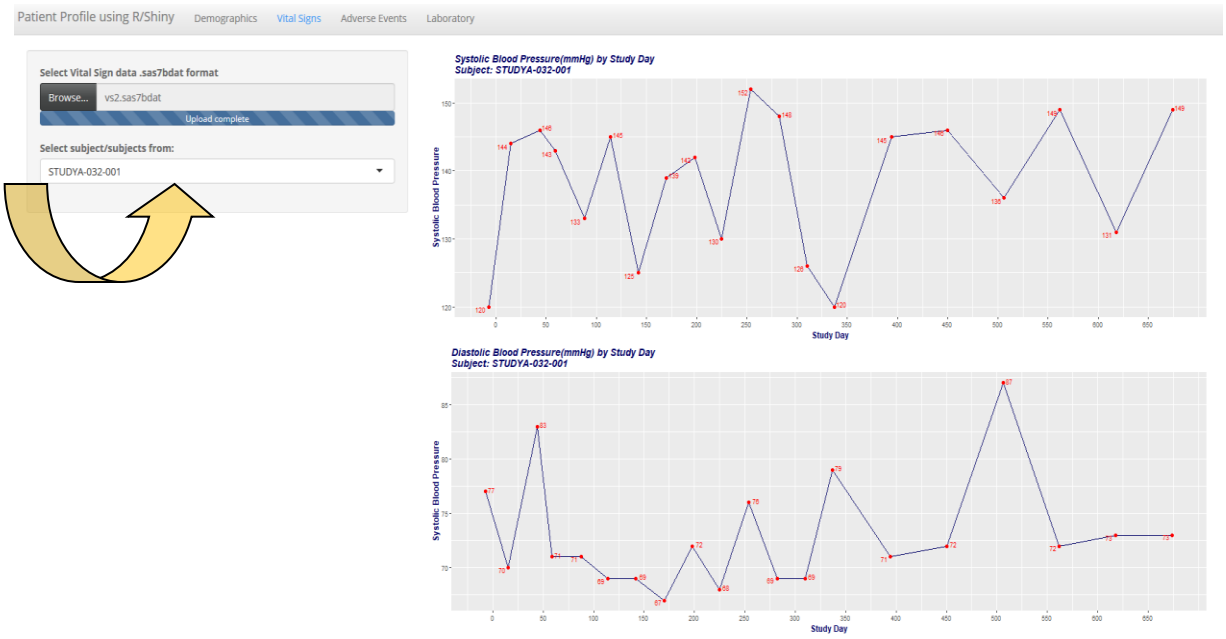
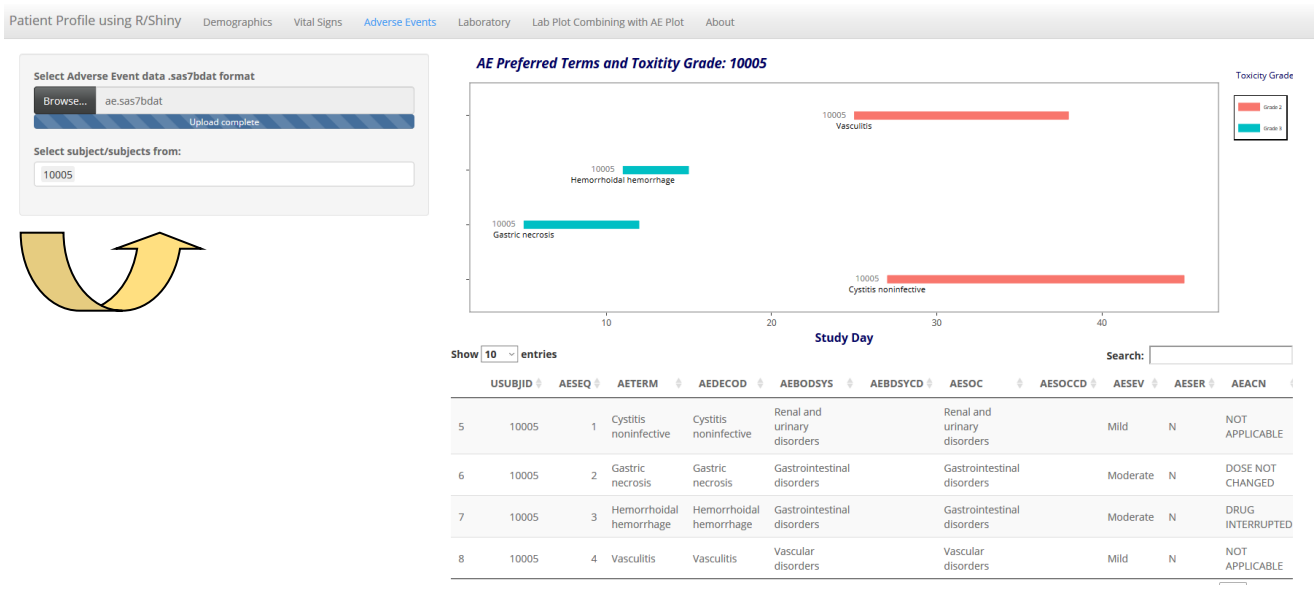


Fig.3

ADVERSE EVENTS, TOXICITY GRADES AND HOVER EFFECT

To present adverse events data, we choose horizontal bars for durations of adverse events over study day for a selected subject as demonstrated in Fig. 4. Toxicity grades of each adverse event are color-coded and each horizontal bar is annotated with the associated adverse event term.



	USUBJID	AESEQ	AETERM	AEDECOD	AEBODSYS	AEBDSYCD	AESOC	AESOCCD	AESEV	AESER	AEACN
5	10005	1	Cystitis noninfective	Cystitis noninfective	Renal and urinary disorders		Renal and urinary disorders		Mild	N	NOT APPLICABLE
6	10005	2	Gastric necrosis	Gastric necrosis	Gastrointestinal disorders		Gastrointestinal disorders		Moderate	N	DOSE NOT CHANGED
7	10005	3	Hemorrhoidal hemorrhage	Hemorrhoidal hemorrhage	Gastrointestinal disorders		Gastrointestinal disorders		Moderate	N	DRUG INTERRUPTED
8	10005	4	Vasculitis	Vasculitis	Vascular disorders		Vascular disorders		Mild	N	NOT APPLICABLE

Fig.4

We add two more functionalities to this adverse event plot. The first functions allows user to select more than one subject as the yellow arrow below points out in Fig. 5. As the second subject is selected, the horizontal bars of adverse events and the data for the second subject are updated automatically .

Fig. 5

The second functionality we add is hover or mouseover effect. This transitional effect can be seen when user moves the mouse over a certain element that makes it slide out, as the blue box shown in Fig. 6. This function is from Shiny library HTMLWIDGET, which is an implementation of the HTML widget JavaScript visualization libraries. It is ingeniously embedded in the RStudio console, so that it is very easy to implement.

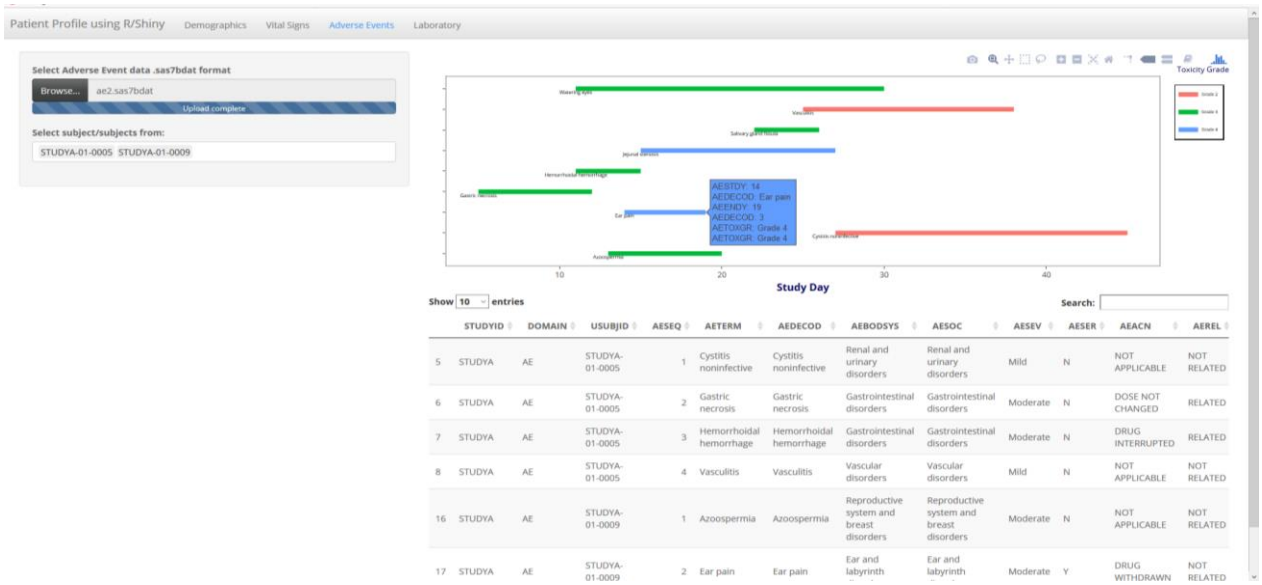


Fig. 6

LAB RESULTS BY LAB TESTS: INTERACTION, INTERACTION AND INTERACTION!

We had two levels of dynamics previously, selecting domain then selecting subject. In the laboratory data example, we add one more level of interactive capability: allowing user to choose different lab tests within a subject. User picks dataset first, then selects a subject, and lastly chooses a laboratory test to display the results as the arrow below points out in Fig. 7.

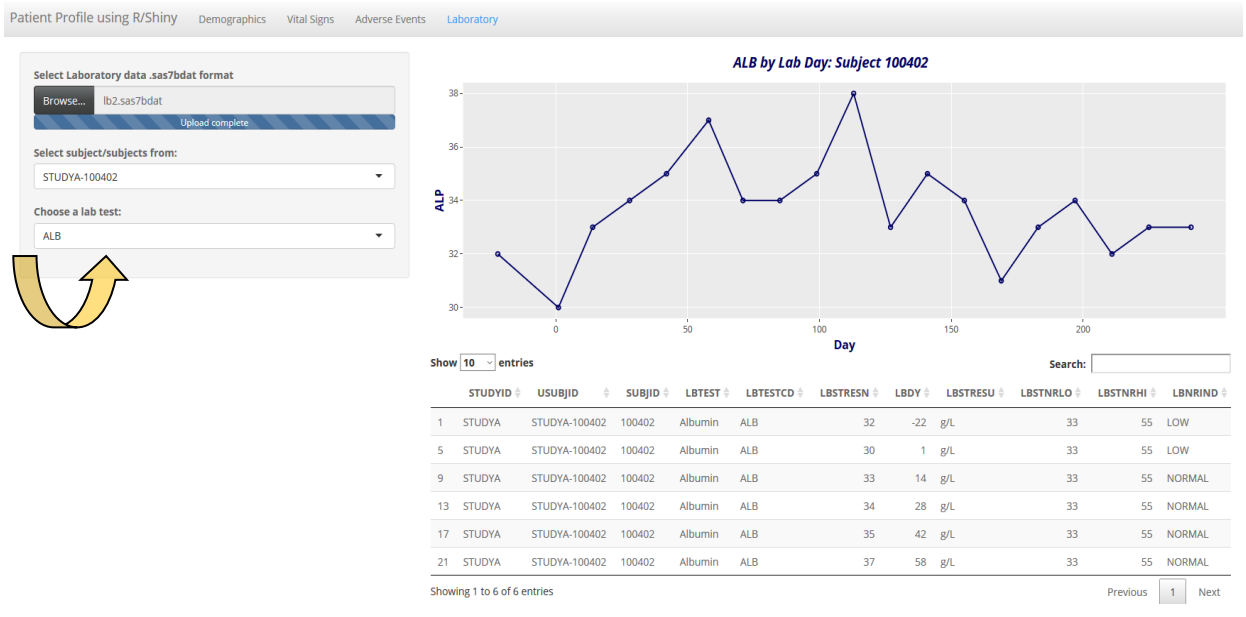


Fig. 7

As subject and laboratory tests are updated, the title of the plots and associated data under the graph will update

dynamically as shown in Fig. 8.

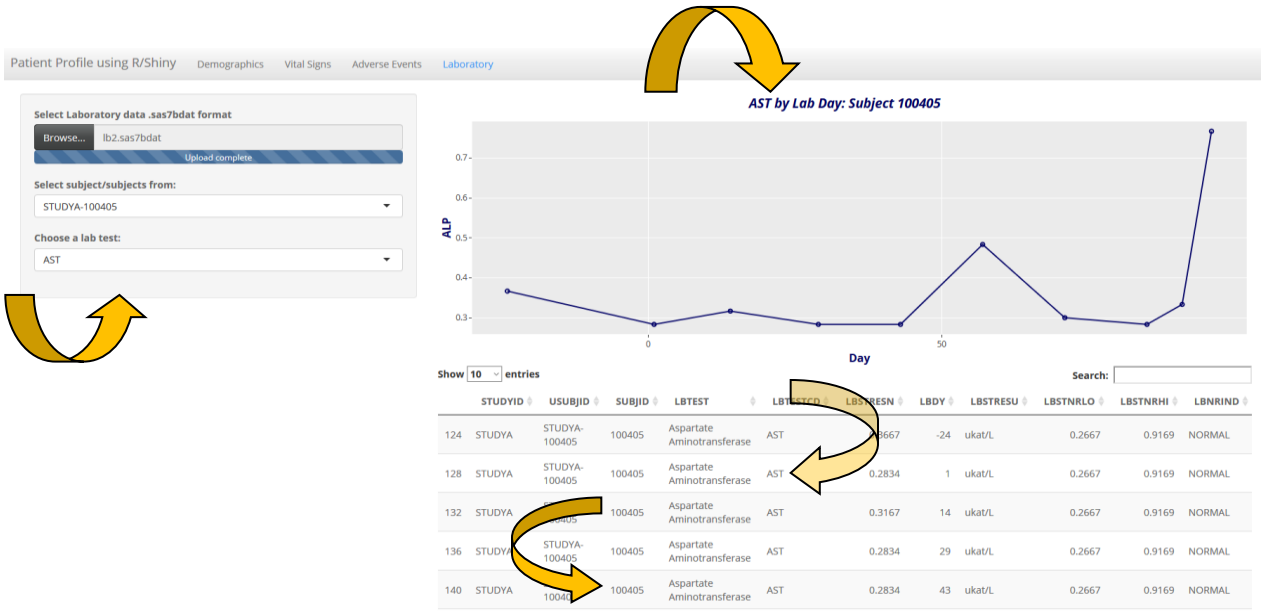


Fig. 8

As user hovers mouse over the plot, the value of the data element will display when HTMLWIDGET function is coded in the program.

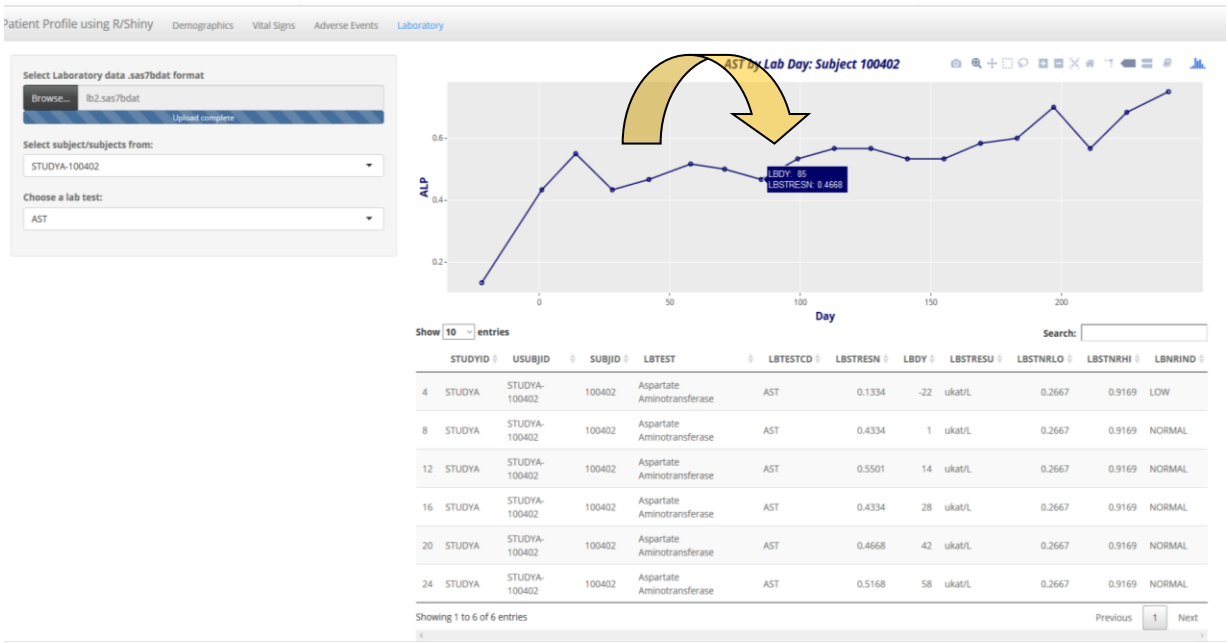


Fig. 9

CONCLUSIONS

Dynamic data visualization, combining multiple plots with data and updating automatically following user's

selections make it easier for reviewers to read/compare data visually, gain insightful information and make best decisions. With great performance and flexibility, Shiny applications can be easily customized to fit different structures of data and provide peculiar designs for different needs. There are numerous topics not covered in this article that can be done with R/Shiny, such as Crosstalk (https://rstudio.github.io/crosstalk/), treemap, 3D or animation (https://shiny.rstudio.com/gallery/).

As the growing amount of health care data continues to accelerate, how we present data to reviewers is crucial. What to present and how to present it can be unclear, the solutions are always best arrived at through in iterative design process incorporating the user requirement feedback. This is a critical difference between open-source R/Shiny tools and other off-the-shelf tools that cannot be further modified. With R/Shiny almost any need of the user can be programmatically implemented, and importantly, implemented in such a manner that the non-technical user is guided through the correct visualizations and analysis tools for their specific tasks.

REFERENCES

https://shiny.rstudio.com/

https://stackoverflow.com/questions/tagged/shiny

https://www.pharmasug.org/proceedings/2018/AD/PharmaSUG-2018-AD24.pdf

CONTACT INFORMATION

Elva Chen
Corvus Pharmaceuticals
863 Mitten Road, Suite 102
Burlingame, CA 94010
Email: echen@corvuspharma.com

Brian Munneke
Corvus Pharmaceuticals
863 Mitten Road, Suite 102
Burlingame, CA 94010
Email: bmunneke@corvuspharma.com