Building Shiny App exercises part 1

INTRODUCTION TO SHINY

Shiny is a package from RStudio that can be used to build interactive web pages with RStudio which is is an open source set of integrated tools designed to help you be more productive with R and you can download it from here. Use the examples in this tutorial to “take a first bite” and prepare for the exercises that follow and will help you build your first Shiny Application from “zero point”. This is the first part of the series and we will just create the interface, make some HTML formatting and add an image to our application. Specifically we will start creating a Shiny Application that will analyze the famous (Fisher’s or Anderson’s) iris data set which gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica. Lets go!

BUILDING INTERFACE (UI)

Every Shiny application includes two parts: a web page which shows the app to the user (UI), and a computer that activates the app (server). You need to create these two parts. UI is just a web document that the user gets to see and is responsible for creating the layout of the app and telling Shiny exactly where things go. The server is responsible for the logic of the app.

You can create a Shiny app by making a new directory and saving a ui.R and server.R file inside it. Each app will need its own unique directory.

You can run a Shiny app by giving the name of its directory to the function runApp(). For example if your Shiny app is in a directory called “Shiny App”, run it with the following code:

```
library(shiny)
runApp("Shiny App")
```

Or by just clicking the “Run App” button at the top of the editor which is the safest solution.

Answers to the exercises are available here.

If you obtained a different (correct) answer than those listed on the solutions page, please feel free to post your answer as a comment on that page.

Learn more about Shiny in the online course R Shiny Interactive Web Apps – Next Level Data Visualization. In this course you will learn how to create advanced Shiny web apps; embed video, pdfs and images; add focus and zooming tools; and many other functionalities (30 lectures, 3hrs.).

Exercise 1

Create a new directory named “Shiny App” in your working directory.

Exercise 2

Create the ui.r and server.r files.

Secondly, we have to install the “Shiny” package with:

```
install.packages("shiny")
```

and then we will call it with:

```
library(shiny)
```

Now that we analyzed the structure of a Shiny app, we will show you how to build a user-interface for this.
To get started, we open the server.R and ui.R files and edit them like this:

```
#ui.R
shinyUI(fluidPage())
#server.R
shinyServer(function(input, output) {})
```

The result is an empty app with a blank user-interface, just to begin with.

Exercise 3

Create an empty app with a blank user-interface.

LAYOUT

Shiny ui.R scripts use the function fluidPage() to create a display that automatically adjusts to the dimensions of your user’s browser window. You lay out your app by placing elements in the fluidPage() function.

For example, the ui.R script below creates a user-interface that has a title panel and then a sidebar layout, which includes a sidebar panel and a main panel. Note that these elements are placed within the fluidPage() function.

```
#ui.R

shinyUI(fluidPage(
titlePanel("title panel"),
sidebarLayout(
sidebarPanel( "sidebar panel"),
mainPanel("main panel"))))
```

The TitlePanel() and sidebarLayout() are the two most used elements to add to fluidPage(). They create a basic Shiny app with a sidebar.

The sidebarLayout() always takes two arguments: sidebarPanel function output and mainPanel function output.

Exercise 4

Create titlePanel(), name it “Shiny App” and sidebarLayout(). Do not forget to add sidebarPanel() and mainPanel() inside this.

HTML Content

You can add content to your Shiny app by placing it inside a *Panel function.

HEADERS

To create a header element: select a header function e.g. (h1) and give it the text you want to see in the header.

For example, you can create a first level header that says “Title” with h1("Title").

To place the element in your app:

Put h1("Title") as an argument to titlePanel(), sidebarPanel(), or mainPanel().

The text will appear in the corresponding panel of your web page. You can place multiple elements in the same panel if you separate them with a comma.

```
#ui.R

shinyUI(fluidPage(
titlePanel("My Shiny App"),
sidebarLayout(
sidebarPanel(),
mainPanel(
h1("First level title"),
h2("Second level title")))))
```

Exercise 5

Create an HTML element to add the title “Menu” in the sidebarPanel() and “Main” in mainPanel() with one of Shiny’s tag functions. HINT: Use h1,h2.

FORMATTED TEXT

Shiny offers many tag functions for formatting text.Take a look:

p: A paragraph of text

h1: A first level header

h2: A second level header

a: A hyper link
br: A line break (e.g. a blank line)
div: A division of text with a uniform style
span: An in-line division of text with a uniform style
pre: Text ‘as is’ in a fixed width font
code: A formatted block of code
img: An image
strong: Bold text
em: Italicized text

Exercise 6

Add a paragraph in your mainPanel() with a description about the app you are going to make.”This famous (Fisher’s or Anderson’s) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.” Use an HTML tag format. HINT:Use p.

Exercise 7

Link the word “iris” in the mainPanel() with this hyperlink “http://stat.ethz.ch/R-manual/R-devel/library/datasets/html/iris.html”. HINT: Use a.

Exercise 8

Add the title “Analysis” under the desciption paragraph of your mainPanel. Do not forget the comma separation. HINT: Use br and h2.

Exercise 9

Use bold text to the words “Iris setosa”,”versicolor” and “virginica”. HINT: Use strong.

IMAGES

Images can improve the appearance of your app and help users understand the content. Shiny uses img() function to put image files in your app. To insert an image, give the img() function the name of your image file as the src argument (e.g., img(src = "my_image.png")). You can also include other HTML parameters such as height and width. For example:

```
img(src = "my_image.png", height = 68, width = 68)
.
```

The img() function looks for your image file in a specific place. Your file must be in a folder named “www” in the same directory as the ui.R script. Shiny will share any file placed here with your user’s web browser, which makes “www” a great place to put images, style sheets, and other things the browser will need to build the web components of your Shiny app. So if you want to use an image named “something.png”, your Shiny App directory should look like this one:

```
#ui.R

shinyUI(fluidPage(
titlePanel("My Shiny App"),
sidebarLayout(
sidebarPanel(),
mainPanel(
img(src="something.png", height = 350, width = 350)))))
```

Exercise 10

Download this image and place it in a folder labeled “www” within your “Shiny App” directory. Name it “petal”, add .jpg extension and then call the img function inside the sidebarPanel(). Use height and width to decide its dimensions.