Bulding Shiny apps

Marlene Marchena

December 7, 2017

R-Ladies Brussels

Material

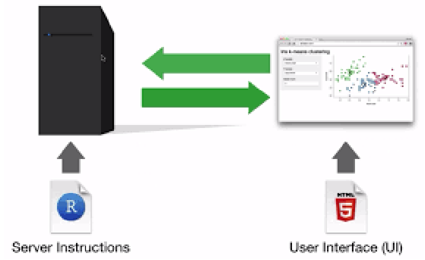
- This presentation is based on the video series: "Hoe to Start Shiny Tutorial" from Rstudio: https://shiny.rstudio.com/tutorial/
- All source code and slides at: https://github.com/rladies/meetup-presentations_brussels
- More references:
  - https://shiny.rstudio.com/gallery/
  - https://www.showmeshiny.com/

What is Shiny ?

- Shiny is an R package, developed by the RStudio, to build interactive web apps straight from R.
- It combines the computational power of R with the interactivity of the web.
- You do not need to know HTML, CSS and JavaScript to deploy a web application.
- A shiny app can help you to present your analytical analysis to a wider audience.

Shiny architecture

- Every shiny app is maintaining by a computer/server running R
- Two main components:
  - User Interface (UI), html instructions
  - Server, R instructions



Basic template

```
library(shiny)

ui <- fluidPage(

)

server <- function(input, output) {

}

shinyApp(ui = ui, server = server)
```

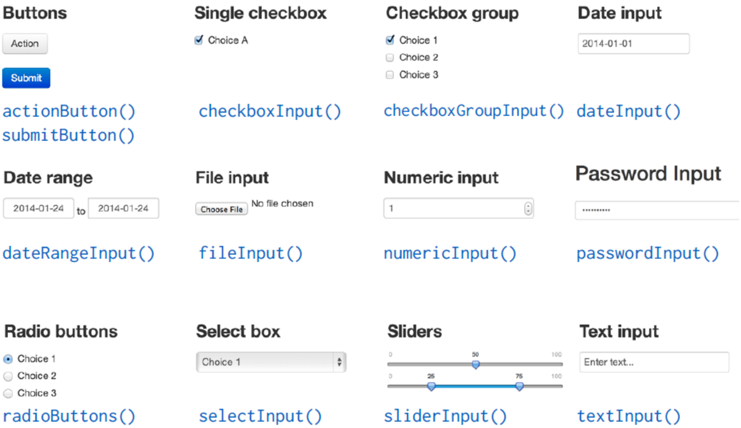
UI

- Control the layout (appearance)
- Create inputs. Syntax example:

  sliderInput(inputId= "num", label= "Choose a number", ..)
- Display outputs. Syntax example:

  plotOutput("hist")

Inputs types



Buttons

Action

Submit

`actionButton()`
`submitButton()`

Single checkbox

☑ Choice A

`checkboxInput()`

Checkbox group

☑ Choice 1
□ Choice 2
□ Choice 3

`checkboxGroupInput()`

Date input

2014-01-01

`dateInput()`

Date range

2014-01-24 to 2014-01-24

`dateRangeInput()`

File input

Choose File No file chosen

`fileInput()`

Numeric input

1

`numericInput()`

Password Input

••••••••••

`passwordInput()`

Radio buttons

● Choice 1
○ Choice 2
○ Choice 3

`radioButtons()`

Select box

Choice 1

`selectInput()`

Sliders

`sliderInput()`

Text input

Enter text...

`textInput()`

Server

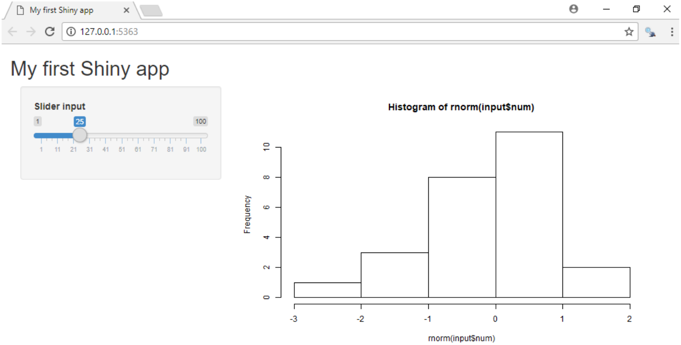
- Access the input value provided by the user using *input$* (Ex. *input$num*)
- Use R functions to process inputs and to produce outputs
- Save the output that you build to *output$* (Ex. *output$hist*)
- Build the output with a *render*()* function

UI and Server working together

Server	UI
renderDataTable()	dataTableOutput()
renderImage()	imageOutput()
renderPlot()	plotOutput()
renderPrint()	verbatimTextOutput()
renderTable()	tableOutput()
renderText()	textOutput()
renderUI()	uiOutput()

Exercise 1

Build the app showed below using sliderInput(inputId = ””,label = ””,value = 25, min = 1, max = 100)



My first Shiny app

127.0.0.1:5363

Slider input

Reactivity

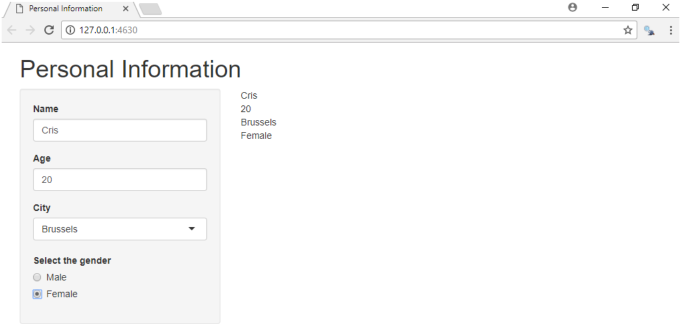
- Reactive values work together with reactive functions.
- Reactive functions build reactivity output to display in UI
- You need a reactive function, render*(), to call a reactive value. For instance:

```
output$hist<-renderPlot({  hist(rnorm(input$num))
```

- Output *hist* will be automatic updated if you change the value of *num*
- Reactivity automatically occurs whenever you use an input value to render an output object

Exercise 2

Build the app showed below:



Personal Information

Name: Cris

Age: 20

City: Brussels

Select the gender

○ Male

● Female

Cris
20
Brussels
Female

Sharing your shiny app

- One directory app.R (your script which ends with a call to shinyApp()). You must use the exact name app.R

- Two file apps: ui.R and server.R in the same working directory

- Using the server maintained by RStudio: shinyapps.io

Thanks for your attention.