

Shiny Apps with R

Subha Srinivasagan

Nutrient Management Spear Program, Cornell University

Overview

1. Shiny Apps - Overview
2. User Interface (UI)
3. Server (Reactive Programming)
4. Adding Image
5. Adding Button
6. Deploying Shiny Apps
7. Resources
8. Example - Simple Calculator

1. R Shiny Overview



- ▶ Rstudio product - Shiny
- ▶ Interactive Web application framework for R
- ▶ Not code heavy - HTML, CSS, and JavaScript
- ▶ Three basic components

- ▶ User-interface

- ▶ Server function

- ▶ A call to Shiny app function

```
# Define user interface for application
ui <- fluidPage(
)

# Define server logic |
server <- function(input, output) {
}

# Run the application
shinyApp(ui = ui, server = server)
```

2. User-interface (UI)

- ▶ Application's front-end appearance
- ▶ Collection of inputs and outputs
 - ▶ Collects inputs from the users
 - ▶ Creates a placeholder to show the outputs
- ▶ Inputs and outputs carry a reference ID that is used in server

2. User-interface (UI)

Basic widgets

Buttons

Action

Submit

`actionButton()`

Single checkbox

☒ Choice A

`checkboxInput()`

Checkbox group

Select all that apply

☒ Choice 1

☐ Choice 2

☐ Choice 3

`checkboxGroupInput()`

Date input

Select date

2014-01-01

`dateInput()`

Date range input

Select dates

2024-01-04 to 2024-01-04

`dateRangeInput()`

File input

Browse... No file selected

`fileInput()`

Help text

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

`helpText()`

Numeric input

Input number

1

`numericInput()`

Radio buttons

Select option

☒ Choice 1

☐ Choice 2

☐ Choice 3

`radioButtons()`

Select box

Select option

Choice 1

`selectInput()`

Sliders

Set value

0 50 100

Set value range

0 25 50 75 100

`sliderInput()`

Text input

Enter text...

`textInput()`

2. User-interface (UI)

```
textInput ("id", "text")
```

```
numericInput("id", "text", value, min, max)
```

Let's create our first web application!

Open file - /Users/home/Desktop/Example1/app.R (GitHub link)

The screenshot shows the RStudio interface with the following components:

- Main Editor:** Displays the file `app.R` with the following content:

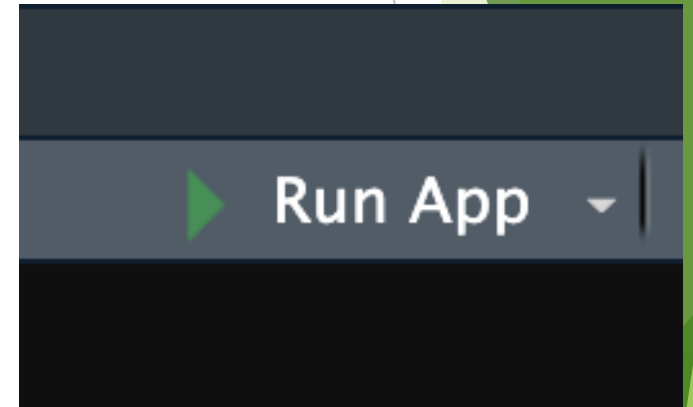
```
1 #####
2 # Shiny using R
3 # Created by Subha Srinivasagan, NMSP, Cornell University
4 # Date: February 17, 2025
5 # This is a Shiny web application. You can run the application by clicking
6 # the 'Run App' button above.
7 # Find out more about building applications with Shiny here:
8 # https://shiny.posit.co/
9 #####
10
11 # Install Shiny package
12 install.packages("shiny")
13
14 #Shiny applications
15 library(shiny)
16
17 # Define user interface for application
18 ui <- fluidPage(
19 )
20
21 # Define server logic
22 server <- function(input, output) {
23 }
24
25 # Run the application
26 shinyApp(ui = ui, server = server)
27
```
- Install Packages Dialog:** A modal window titled "Install Packages" is open. It shows:
 - Install from:** Repository (CRAN)
 - Packages (separate multiple with space or comma):** shiny
 - A list of suggested packages: shiny, shiny.benchmark, shiny.blueprint, shiny.destroy, shiny.emptystate, shiny.exe, shiny.fluent, shiny.i18n, shiny.info, shiny.ollama, shiny.pwa, shiny.react, shiny.reglog, shiny.router, shiny.semantic, shiny.tailwind, shiny.telemetry, shiny2docker, shinyAce, shinyaframe, shinyalert.
 - Buttons:** Install (highlighted with a red box), Cancel.
- Right Sidebar:** The "Packages" tab is active, showing a list of installed and available packages. The "Install" button in the top bar is also highlighted with a red box.
- Console:** Shows the R version (R 4.4.1) and the current directory (~/.Desktop/Test1/).

2. User-interface (UI)

```
textInput ("id", "text")  
numericInput("id", "text", value, min, max)
```

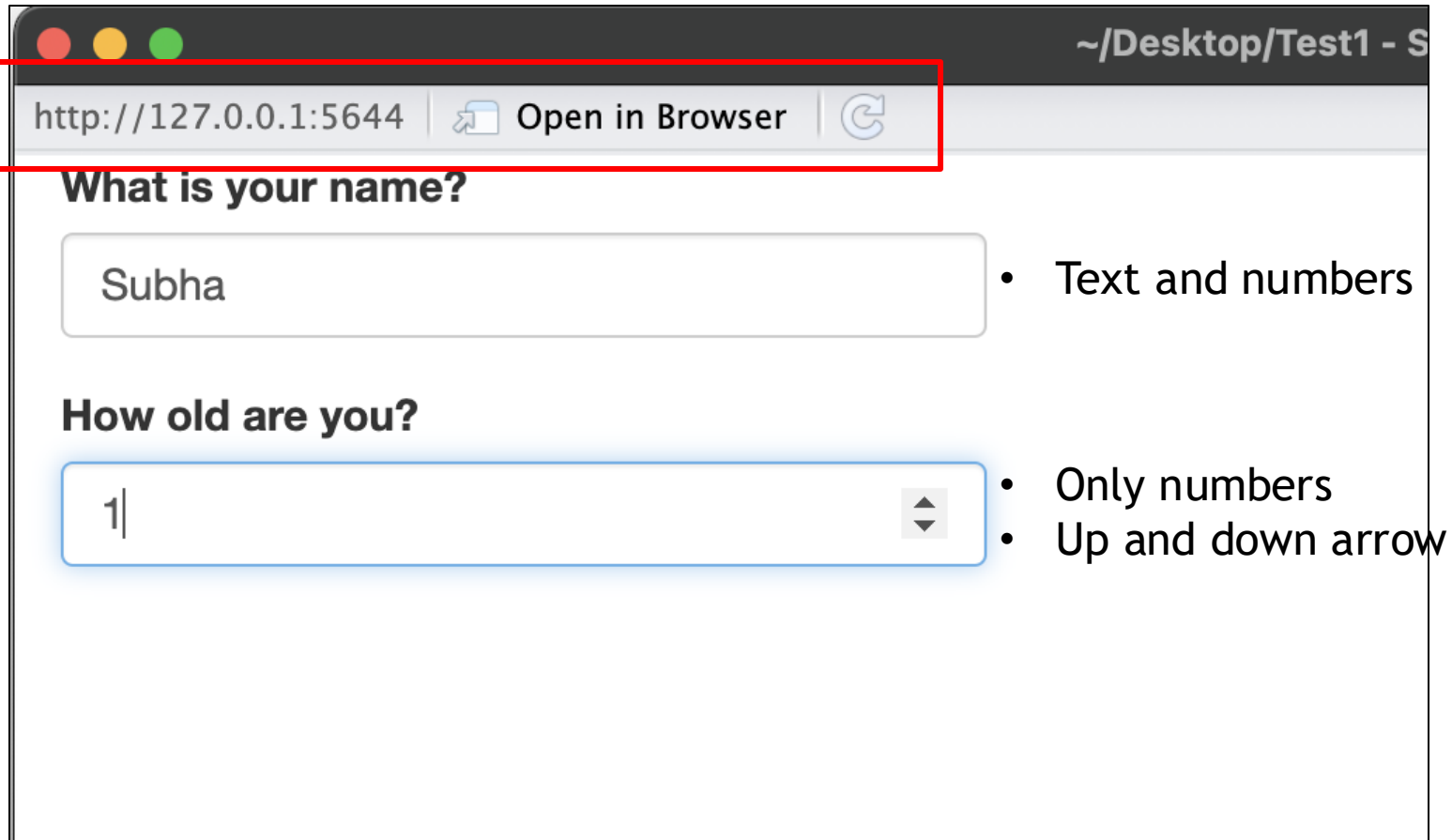
```
# Define user interface for application  
ui <- fluidPage(  
  
  # Taking inputs  
  textInput("name", "What is your name?"),  
  numericInput("age", "How old are you?", value = NULL, min = 0),  
)
```

Don't forget!



2. User-interface (UI)

Local server



The screenshot shows a web browser window with the address bar containing `http://127.0.0.1:5644`. The browser title is `~/Desktop/Test1 - S`. The page content includes two form fields:

- What is your name?**
A text input field containing the text "Subha".
 - Text and numbers
- How old are you?**
A number input field containing the text "1".
 - Only numbers
 - Up and down arrow

3. Server (Reactive Programming)

- Application's back end logic
- Creates and renders output based on inputs from UI
- Directly render or react to actions
- Render functions should match the input functions

3. Server (Reactive Programming)

```
# Define user interface for application
ui <- fluidPage(

  # Taking inputs
  textInput("name", "What is your name?"),
  numericInput("age", "How old are you?", value = NULL, min = 0),

  # Placeholder for output
  textOutput("text"),

) id
```

```
# Define server logic
server <- function(input, output) {

  output$text <- renderText("Hello World! My name is")

}
```

What is your name?

Subha

How old are you?

Hello World! My name is

3. Server (Reactive Programming)

```
# Define server logic
server <- function(input, output) {

  output$text <- renderText(paste("Hello World! My name is",input$name))

}
```

`paste("string",variable)`

What is your name?

Subha

How old are you?

1

Hello World! My name is Subha

```
# Taking inputs
textInput("name", "What is your name?"),
      id
```

3. Server (Reactive Programming)

What is your name?

Subha

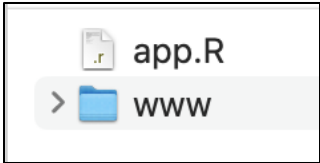
How old are you?

1

Hello World, my name is Subha and I am 1 years old.

4. Adding Image

Images are required to be stored in a folder names www




```
# Define user interface for application
ui <- fluidPage(

  # Displaying image
  img(src = "NMSP.png", height = "100px", width = "100px"),

  # Taking inputs
  textInput("name", "What is your name?"),
  numericInput("age", "How old are you?", value = NULL, min = 0),

  # Placeholder for output
  textOutput("text")

)
```



What is your name?

How old are you?

Hello World, my name is Subha and I am 1 years old.

5. Adding Button

Invokes the reactivity action (not just render) from the server

Buttons

Action

Submit

actionButton()

```
# Define user interface for application
ui <- fluidPage(

  # Displaying image
  img(src = "NMSP.png", height = "100px", width = "100px"),

  # Taking inputs
  textInput("name", "What is your name?"),
  numericInput("age", "How old are you?", value = NULL, min = 0),

  # Action button to trigger text display
  actionButton("show_text", "Show Text"),

  # Placeholder for output
  textOutput("text")

)
```

5. Adding Button

Invokes the reactivity action (not just render) from the server

```
# Define server logic
server <- function(input, output) {

  #Reactive value to store button click event
  text_reactive <- eventReactive(input$show_text, {
    paste("Hello World, my name is", input$name, "and I am", input$age, "years old.")
  })

  output$text <- renderText(text_reactive())

}
```



What is your name?

How old are you?

5. Adding Button

Invokes the reactivity action (not just render) from the server

```
# Define server logic
server <- function(input, output) {

  #Reactive value to store button click event
  text_reactive <- eventReactive(input$show_text, {
    paste("Hello World, my name is", input$name, "and I am", input$age, "years old.")
  })

  output$text <- renderText(text_reactive())

}
```



What is your name?

How old are you?

Show Text

Hello World, my name is Subha and I am 1 years old.

6. Deploying Shiny Apps

1. Run locally - local host server

- ▶ Run App button

- ▶ `shiny::runApp('app', port = 3838)`

2. Web-based hosting

- ▶ Shinyapps.io

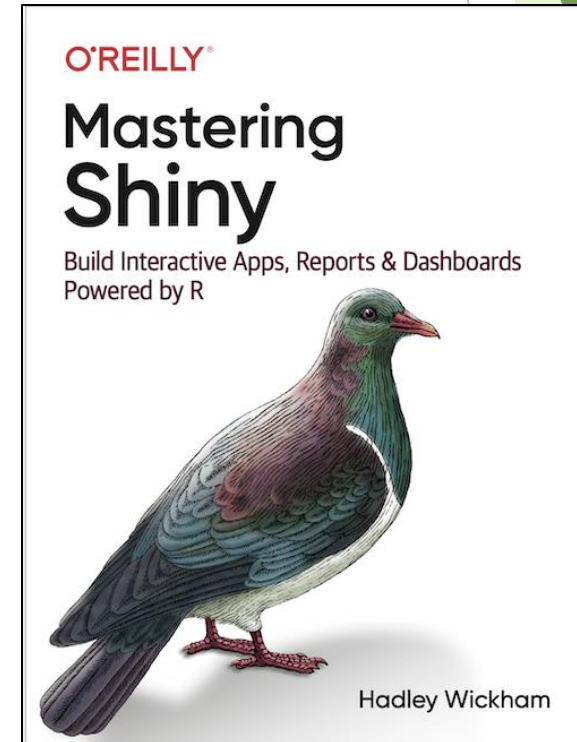
- ▶ Rstudio Connect

- ▶ Shiny server

- ▶ Heroku

7. Resources

1. Rstudio tutorial - [link](#)
2. Mastering Shiny by Hadley Wickham - [link](#)
 - ▶ Free online book



8. Calculator Webtool Using Shiny in R

Simple Calculator

Created by Subha Srinivasagan, NMSP, Cornell University



Enter first number:

Enter second number:

Addition (+)

Subtraction (-)

Multiplication (×)

Division (÷)

Thank you!



sn558@cornell.edu



<https://github.com/Subhashree9>