09\_shiny

Richard Bui

2023

## What is Shiny?

Shiny is a web-app using R programming. In other words, it allows us to build an interactive app that may useful for data analysis. For example:

* Pick a variable from a dataset then draw its histogram or calculate its statistics
* Pick two variables and run a regression between two variables, then show us all the coefficients

## Why do we need a Shiny app?

* For users!
  + They don’t know R, say, your boss or your parents (most of them)
  + To sell your products later

## Components of a Shiny app

* You need to install and use the package shiny

library(shiny)  
library(tidyverse)

-- Attaching packages --------------------------------------- tidyverse 1.3.1 --

v ggplot2 3.3.6 v purrr 0.3.4  
v tibble 3.1.7 v dplyr 1.0.9  
v tidyr 1.2.0 v stringr 1.4.0  
v readr 1.4.0 v forcats 0.5.1

* An app has two parts:
  1. ui (user interface, front end): you design an interface for users to use your app
  2. serve (backend): to receive the request from user, analyze it, get the output, and return to the user at front end
* Then, we combine them together via a function like shinyApp(ui, server)

## Example 1: Statistics of one variable

* Assume we have a dataset already, e.g., our student score dataset

Hsb <- within(  
 read.csv("https://stats.idre.ucla.edu/stat/data/hsb2.csv"), {  
 race <- as.factor(race)  
 schtyp <- as.factor(schtyp)  
 prog <- as.factor(prog)  
})

* Take a look at data:

head(Hsb)

id female race ses schtyp prog read write math science socst  
1 70 0 4 1 1 1 57 52 41 47 57  
2 121 1 4 2 1 3 68 59 53 63 61  
3 86 0 4 3 1 1 44 33 54 58 31  
4 141 0 4 3 1 3 63 44 47 53 56  
5 172 0 4 2 1 2 47 52 57 53 61  
6 113 0 4 2 1 2 44 52 51 63 61

* The students may have scores in 4 subjects: reading, writing, math, and science. Our purpose/problem is to allow users to understand the statistics of these 4 scores.
  + **[Front end]:** The user can pick any subject at the front end
  + **[Back end]:** After the user chose the variable, we select that variable from the data Hsb
  + **[Back end]:** We analyze that variable, e.g., simple take summary function to get the statistics of the variable
  + **[Front end]:** Return the output to the user at the front end

## Part 1: ui

* Let’s design our ui: we need two things
  + **[Front end, step 1]:** The user can pick any subject at the front end
  + **[Front end, step 2]:** Return the output to the user at the front end
* It requires your imagination, creativity, and learn from other apps. Now, ui design is a skill and a potential career for everyone
* In class, I will not go to details how to make it beautiful. You need to learn yourselves. I just give you very simple design in the class (I have poor creativity!)
* OK. We have two parts of the ui:
* So many possible layouts to choose to pick a thing in Shiny
* Please see this website to learn more: <https://shiny.rstudio.com/articles/layout-guide.html>
* How about we divide the app’s **layout** into two areas (**sidebar layout**):
  + Left side: allow the user to pick the variable
  + Right side: show the output after we analyzed data at the front end

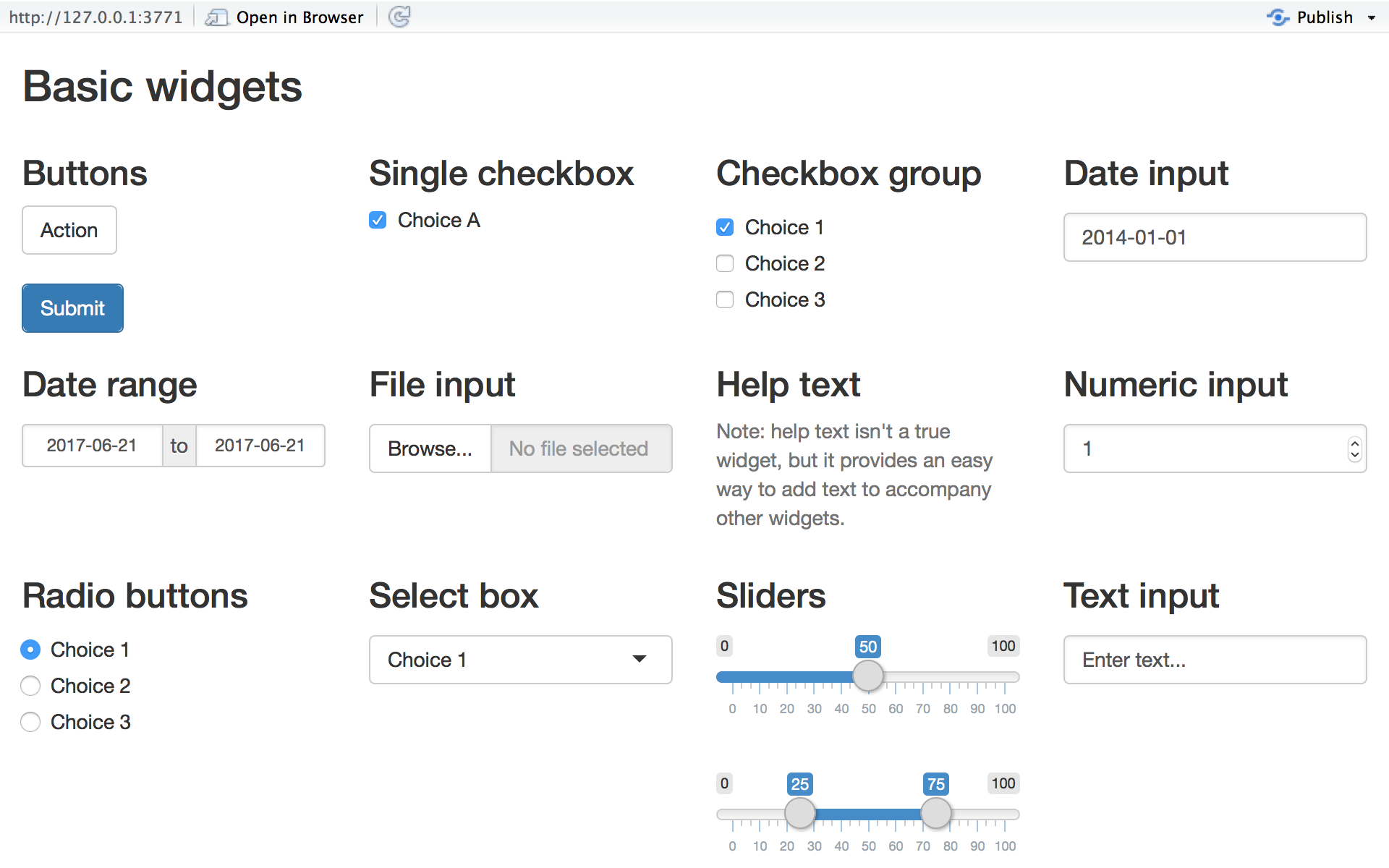
## Sidebar layout: to split the ui into two areas

* It will look like this:

ui <- fluidPage(  
  
 # App title ----  
 titlePanel("Shiny First App"),  
  
 # Sidebar layout  
 sidebarLayout(  
  
 # LEFT: Sidebar panel for inputs ----  
 sidebarPanel(  
 # Input: WRITE YOUR CODE HERE (1)  
  
 ),  
  
 # RIGHT: Main panel for displaying outputs ----  
 mainPanel(  
 # Output: WRITE YOUR CODE HERE (2)  
 )  
 )  
)

* Because you haven’t written your code so nothing appears

## Input: select widget

* Now, let’s write the code for input
* To pick a variable: we need to have a control widget to pick the variable
* Read more on widget at here: <https://shiny.rstudio.com/tutorial/written-tutorial/lesson3/> and the gallery of all possible widgets <http://shiny.rstudio.com/gallery/widget-gallery.html>
* Again, there are so many widgets to choose so that we can pick something in Shiny:
* 
* For our problem, we want to design so that the user can pick one of these 4 possible choices: read, write, math, and science
* Which widget you think will fit your problem?
  + Buttons: not for picking things
  + Text input: the user doesn’t know the variable names, typo, lazy users
  + Date: for dates picking only
  + Sliders: more suitable for picking numeric ranges
  + Select box may be good
* OK. Let’s use select box widget: selectInput with four possible choices

sidebarPanel(  
 # Input: WRITE YOUR CODE HERE (1)  
 selectInput("var", # name of this widget  
 h3("Select one variable"), # label of this widget  
 choices = list("read" = "read", # "user select this" = value we set for this option, will be used later in the back end  
 "write" = "write",  
 "math" = "math",  
 "science" = "science"  
 ),   
 selected = "read" # default of the choice in app  
 )  
)

## Output: add R object to ui

* After analyzing data, we need to return the output (an R object) to the ui’s main panel
* The R objects could be the following:
  + text: textOutput
  + table: tableOutput
  + plot: plotOutput
  + image: imageOutput
  + Others: please read more at here <https://shiny.rstudio.com/tutorial/written-tutorial/lesson4/>
* For example, the below code is to just simply write a text to the output panel:

mainPanel(  
 textOutput("selected\_var") # selected\_var is the name of the output, that we will create later in the server (back end)  
 )

## The final ui

ui <- fluidPage(  
   
 # App title ----  
 titlePanel("Shiny First App"),  
   
 # Sidebar layout  
 sidebarLayout(  
   
 # LEFT: Sidebar panel for inputs ----  
 sidebarPanel(  
 # Input: WRITE YOUR CODE HERE (1)  
 selectInput("var", # name of this widget  
 h3("Select one variable"), # label of this widget  
 choices = list("read" = "read", # "user select this" = value we set for this option, will be used later in the back end  
 "write" = "write",  
 "math" = "math",  
 "science" = "science"  
 ),   
 selected = "read" # default of the choice in app  
 )  
 ),  
   
 # RIGHT: Main panel for displaying outputs ----  
 mainPanel(  
 # Output: WRITE YOUR CODE HERE (2)  
 textOutput("selected\_var")  
 )  
 )  
)

## Part 2: server

* The idea of server is to link between input-analyzing-output
* For example, it looks like this:

server <- function(input, output) {  
   
 output$selected\_var <- renderText({   
 paste0("You have selected the ", input$var, " variable") # var is the name of widget from the input sidebar, that the user picked it  
 })  
   
}

* Depending on the output, we can render in the server by different render functions:
  + Text: renderText
  + Table: renderTable
  + Plot: renderPlot
  + Others: please read more at here <https://shiny.rstudio.com/tutorial/written-tutorial/lesson4/>

## Launch the app

myapp = shinyApp(ui, server)  
runApp(myapp)

## More analysis at the server

* The app worked like a charm, but it is a bit quite stupid :)
* Now, let’s make more analysis for the variable the user picked, then get the output, and render to the ui’s main panel

server <- function(input, output) {  
   
 output$selected\_var <- renderText({   
 paste0("You have selected the ", input$var, " variable. Below is its statistics: ") # var is the name of widget from the input sidebar, that the user picked it  
 })  
   
 # MORE CODE FOR ANALYSIS  
 output$stat\_var = renderText(  
 summary(Hsb %>% select(input$var))  
 )  
}

* Show the output to the main panel

ui <- fluidPage(  
   
 # App title ----  
 titlePanel("Shiny First App"),  
   
 # Sidebar layout  
 sidebarLayout(  
   
 # LEFT: Sidebar panel for inputs ----  
 sidebarPanel(  
 # Input: WRITE YOUR CODE HERE (1)  
 selectInput("var", # name of this widget  
 h3("Select one variable"), # label of this widget  
 choices = list("read" = "read", # "user select this" = value we set for this option, will be used later in the back end  
 "write" = "write",  
 "math" = "math",  
 "science" = "science"  
 ),   
 selected = "read" # default of the choice in app  
 )  
 ),  
   
 # RIGHT: Main panel for displaying outputs ----  
 mainPanel(  
 # Output: WRITE YOUR CODE HERE (2)  
 textOutput("selected\_var"),  
 textOutput("stat\_var")  
 )  
 )  
)

* Let’s launch the app again:

myapp = shinyApp(ui, server)  
runApp(myapp)

* A bit ugly, but worked!
* Let’s add a histogram of the chosen variable into the main panel: need to revise both ui and server to include it:

ui <- fluidPage(  
   
 # App title ----  
 titlePanel("Shiny First App"),  
   
 # Sidebar layout  
 sidebarLayout(  
   
 # LEFT: Sidebar panel for inputs ----  
 sidebarPanel(  
 # Input: WRITE YOUR CODE HERE (1)  
 selectInput("var", # name of this widget  
 h3("Select one variable"), # label of this widget  
 choices = list("read" = "read", # "user select this" = value we set for this option, will be used later in the back end  
 "write" = "write",  
 "math" = "math",  
 "science" = "science"  
 ),   
 selected = "read" # default of the choice in app  
 )  
 ),  
   
 # RIGHT: Main panel for displaying outputs ----  
 mainPanel(  
 # Output: WRITE YOUR CODE HERE (2)  
 textOutput("selected\_var"),  
 textOutput("stat\_var"),  
 plotOutput("hist\_var")  
 )  
 )  
)

server <- function(input, output) {  
   
 output$selected\_var <- renderText({   
 paste0("You have selected the ", input$var, " variable") # var is the name of widget from the input sidebar, that the user picked it  
 })  
   
 # MORE CODE FOR ANALYSIS  
 output$stat\_var = renderText(  
 summary(Hsb %>% select(input$var))  
 )  
   
 output$hist\_var = renderPlot(  
 hist(Hsb %>% pull(input$var), # pull to get the vector, rather than dataframe   
 main = paste0("Histogram plot of ", input$var)  
 )   
 )  
}

* Run the app again:

myapp = shinyApp(ui, server)  
runApp(myapp)

## Practice quiz

* Simple one: I want to allow the user to have 5 possible choices for variables: read, write, math, science, and socst . Could you help to revise the app to have 5 possible choices for input.
* Medium one: I want to allow the user to pick two variables (each can have 4 possible choices: read, write, math, science). Then, plot the scatter plot between two variables.
* Medium one: Render the head of the table Hsb by showing only the first 6 rows of the table to the main panel
* Difficult one:
  + Allow the user to have a slider bar (range from 0 to 100) to choose one threshold value, default value is 50
  + Then, in the server, run a t-test one sample to compare mean of the chosen variable and the threshold
  + Then, get the output and print out to the main panel of the ui
* Have fun with both quiz