Post02_Anna_Lu.Rmd

Anna_Lu

Dec, 2, 2017

Shinydashboard – Quickly create professional shiny app

Installation and library()

shinydashboard requires Shiny 0.11 or above. There is the way to install packages.

```
#install.packages("shiny")
#install.packages("shinydashboard")
#install.packages("ggplot2")
library(shiny)
library(shinydashboard)
```

```
##
## Attaching package: 'shinydashboard'
```

```
## The following object is masked from 'package:graphics':
##
##     box
```

```
library(ggplot2)
```

1. Introduction

First, I would like to introduce Shiny because some of you might be not quite familiar with Shiny.

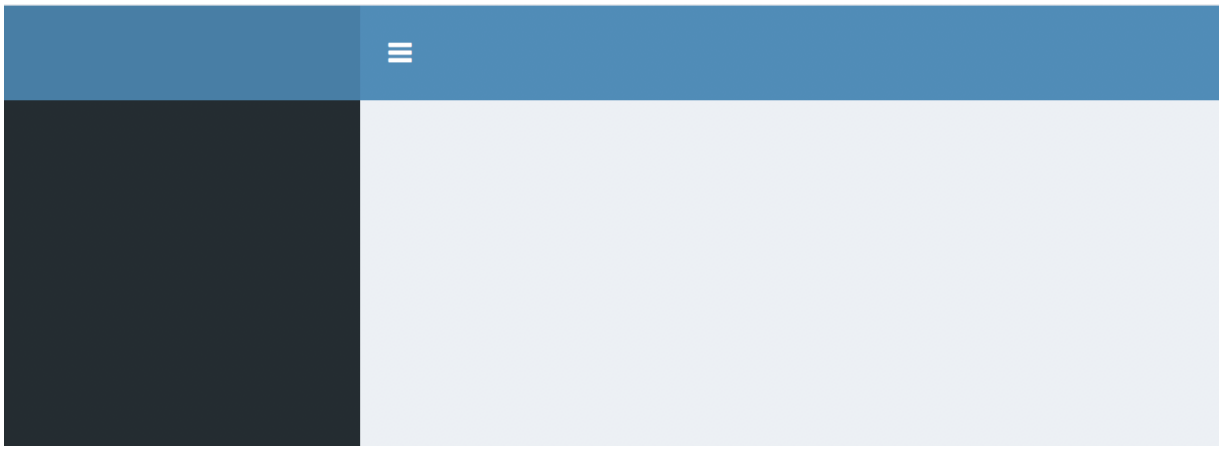
“Shiny is an R package that makes us easy to build interactive web apps straight from R.(shiny.rstudio)” In RStudio website, they define Shiny using “Interact, Analyze and Communicate”. Using Shiny package, we can just use simpler code to build grade web app. Also, Shiny is flexible. It can refresh real-time data, it accpet R, HTML, SCC and JavaScript and Shiny has lots of useful tools, like slider widget.

Compared with other language, R shiny package is easier to learn. More exciting, shinydashboard package makes Shiny even better and more handy. The shinydashboard uses Shiny UI code and it is dynamic. Using the shinydashboard, we can quickly create a professional dashboard.

2. shinydashboard structure

There are three components of shinydashboard, header, sidebar and body. Header contains the title and a dropdown menus. Siderbar is a place to pust some wedgets. We can have slider, checkbox and so on. It is like a main console. The last part is body. If We call siderbar is a main console, body is like main display console. Okay, we already know some basic information of shiny and shinydashboard. Right now it is time to see the real code. WOW, Excited ! (ﾉ ✧∇✧)ﾉ

```
#1) Simplest whole structure output in the world

ui <- dashboardPage(
  dashboardHeader(),
  dashboardSidebar(),
  dashboardBody()
)

server <- function(input, output) { }

shinyApp(ui, server)
```



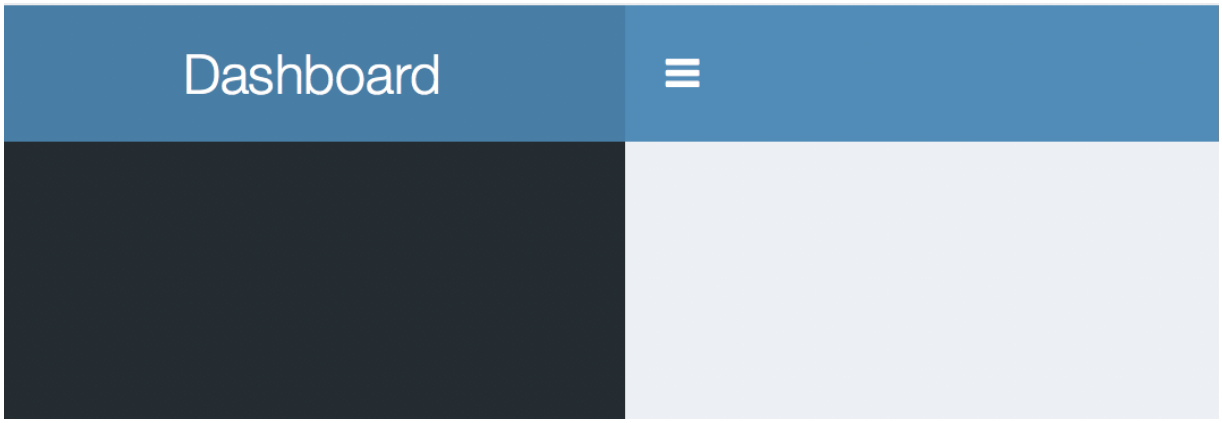
structure

```
#2) When we want to creat more complicated app, this structure is very helful. The output is as same as the previo
us one

header <- dashboardHeader()
sidebar <- dashboardSidebar()
body <- dashboardBody()

dashboardPage(header, sidebar, body)

server <- function(input, output) { }
shinyApp(ui, server)
```

3. Header

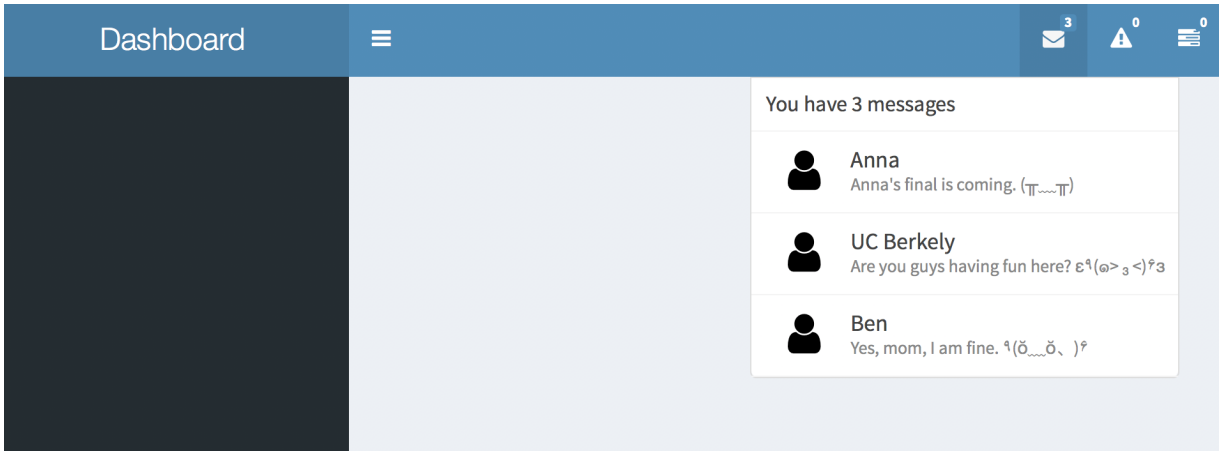
After knowing how to write the simplest code of shinydashboard, we can start to learn how to fill in the blank. Let's start with header

```
#1) Using 'title =' to get a title

ui <- dashboardPage(
  dashboardHeader(title = "Dashboard"),
  dashboardSidebar(),
  dashboardBody()
)

server <- function(input, output) { }
shinyApp(ui, server)
```



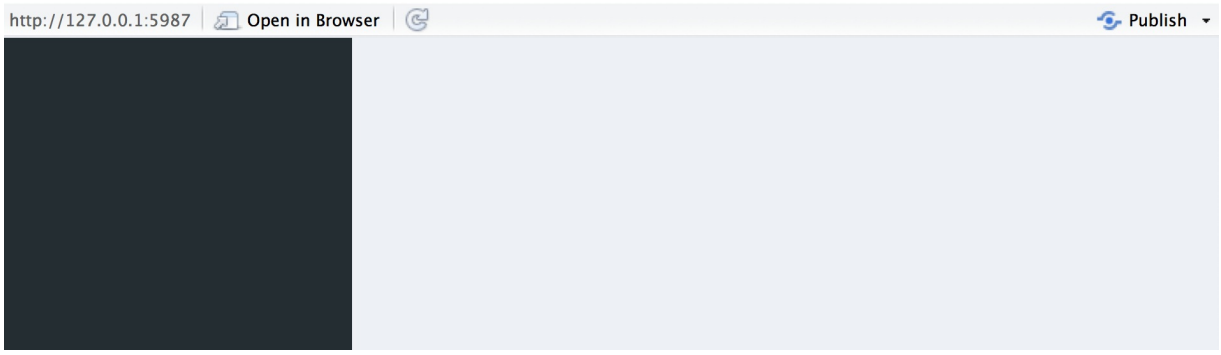
header

```
#2) We can have a dropdown menus by using 'dropdownMenu()'

ui <- dashboardPage(
  dashboardHeader(title = "Dashboard",
    dropdownMenu(type = "messages",
                 messageItem(
                   from = "Anna",
                   message = "Anna's final is coming. (╥﹏╥) "
                 ),
                 messageItem(
                   from = "UC Berkely",
                   message = "Are you guys having fun here? ε٩(๑> ₃ <)۶з "
                 ),
                 messageItem(
                   from = "Ben",
                   message = "Yes, mom, I am fine. ٩(ŏ﹏ŏ、 )۶ "
                 )),
    dropdownMenu(type = "notification"),
    dropdownMenu(type = "task")
  ),
  dashboardSidebar(),
  dashboardBody()

)

server <- function(input, output) { }
shinyApp(ui, server)
```



dropdownmenu

```
#3) If we dont want the header, we can use 'disable = TRUE'
ui <- dashboardPage(
  dashboardHeader(title = "Dashboard", disable = TRUE),
  dashboardSidebar(),
  dashboardBody()
)

server <- function(input, output) { }
shinyApp(ui, server)
```

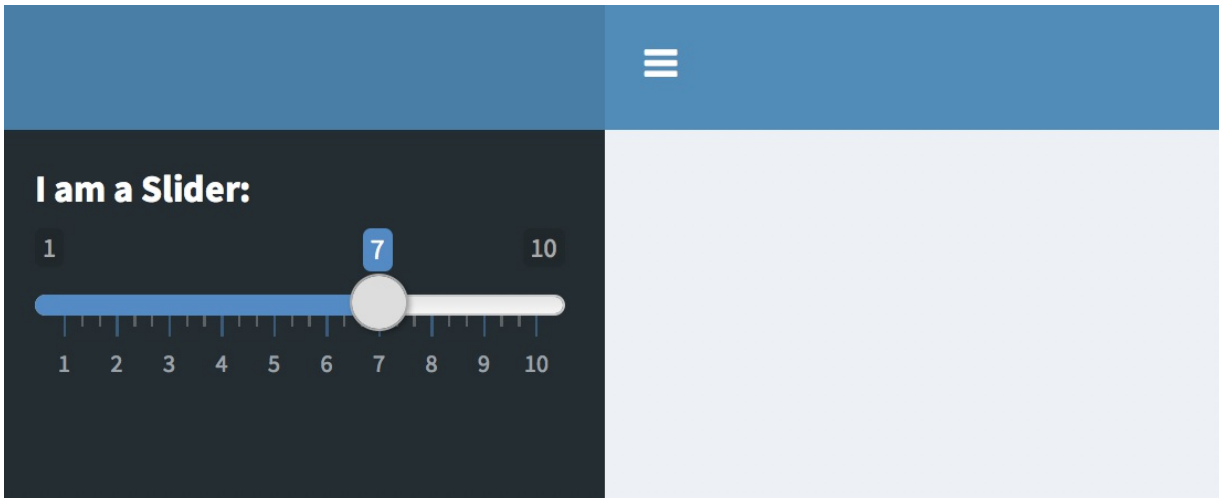


disable

4. Sidebar

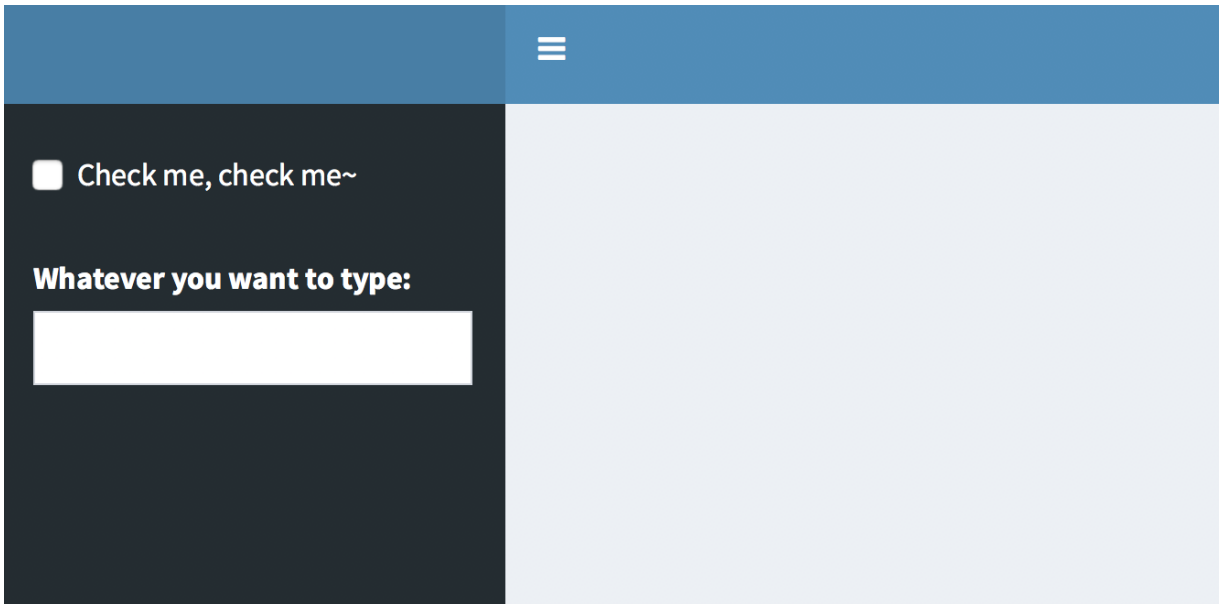
Sidebar is my favourite part, and it is most important part. Like people's brain, Sidebar control the dashboard. There are many widgets of shinydashboard. Right now, I just talk about some popular tools. Checkbox, text and Slider.

```
#1) Slider, using 'sliderInputs()'

ui <- dashboardPage(
  dashboardHeader( ),
  dashboardSidebar(
    sliderInput("slider", label = "I am a Slider:", min = 1, max = 10, step = 0.5, value = 7)
  ),
  dashboardBody( )
)

server <- function(input, output) { }
shinyApp(ui, server)
```



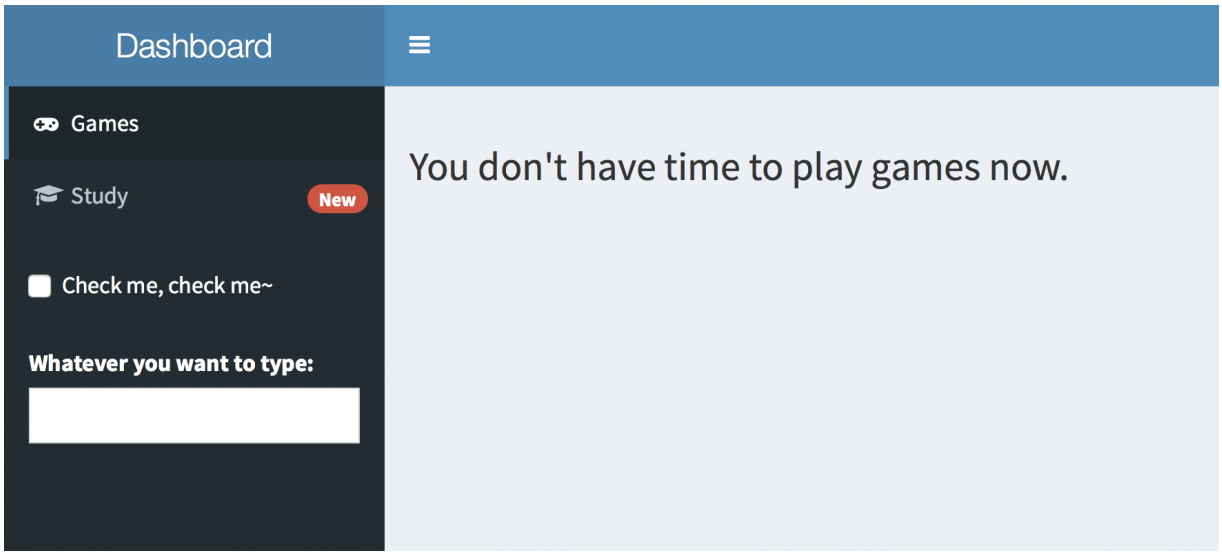
slider

```
#2) We can add more widgets. Checkbox, using 'checkboxInput()'. Text input, using 'textInput()'

ui <- dashboardPage(
  dashboardHeader( ),
  dashboardSidebar(
    checkboxInput("check", label = "Check me, check me~"),
    textInput("text", label = "Whatever you want to type: ")
  ),
  dashboardBody( )
)

server <- function(input, output) { }
shinyApp(ui, server)
```



Sidebar

```
#3) For sidebar, we have the tabs in the menu items, which is similar to tabPanel from Shiny. In Shiny base, we ne
ed to use conditionalPanel, which is much more confused. In shinydashboard, 'tabItem()' and 'menuItem' really ligh
ten our workload. There is also 'tabBox()' for the body part. I will talk about it later.

  sidebar <- dashboardSidebar(
    sidebarMenu(
      menuItem("Games", tabName = "games", icon = icon("gamepad")),
      menuItem("Study", tabName = "study", icon = icon("graduation-cap"),
               badgeLabel = "New", badgeColor = "red")
    ),
    checkboxInput("check", label = "Check me, check me~"),
    textInput("text", label = "Whatever you want to type: ")
  )

  body <- dashboardBody(
    tabItems(
      tabItem(tabName = "study",
              h3("You still have 3 Final Projects due.")),
      tabItem(tabName = "games",
              h3("You don't have time to play games now."))

  )
)

ui <- dashboardPage(
  dashboardHeader(title = "Dashboard"),
  sidebar,
  body
)

server <- function(input, output) { }
shinyApp(ui, server)
```

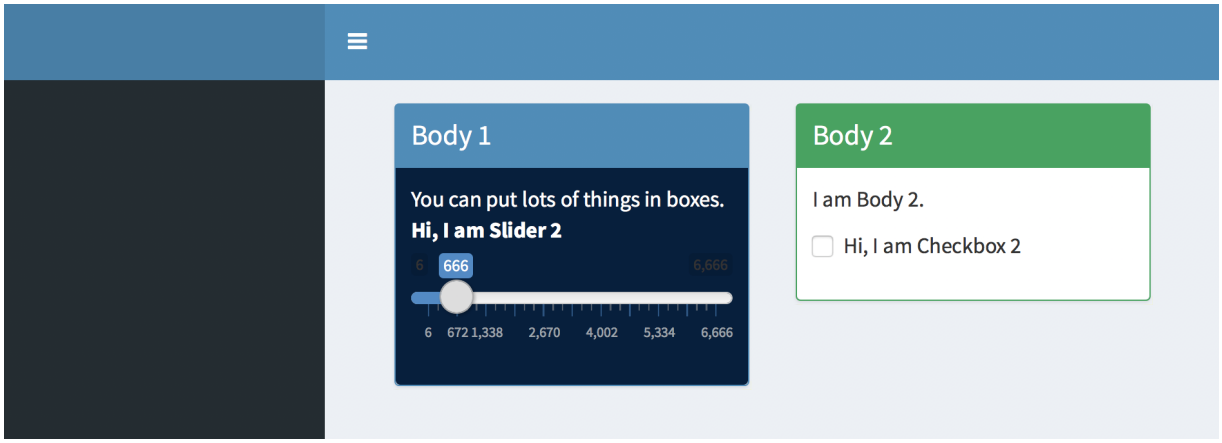


menu, tab

5. Body

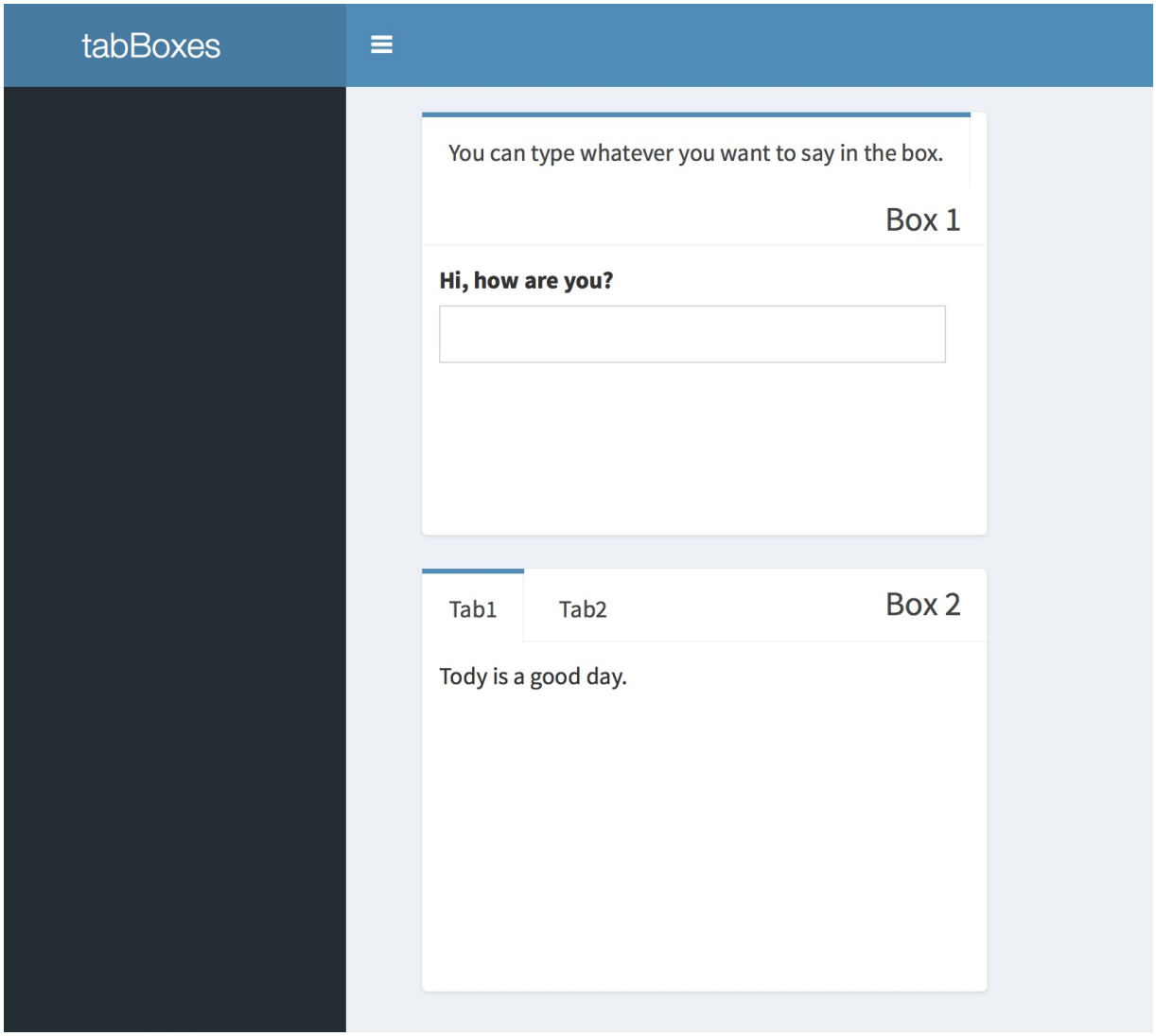
The body is like the body paragraph of the essay. This is the place to contain the Shiny content. Usually, we will use Boxs to contain the content.

```
#1) Typically, the boxes will be in 'fluidRow()'. In the 'box()', we have title, status and some other options.

ui <- dashboardPage(
  dashboardHeader(),
  dashboardSidebar(),
  dashboardBody(
    fluidPage(
      box(
        title = 'Body 1', status = 'primary', solidHeader = TRUE,
        background = "navy",
        "You can put lots of things in boxes.",
        sliderInput("slider", "Hi, I am Slider 2", min = 6, max = 6666, value = 666)
       ),
      box(
        title = 'Body 2', status = 'success', solidHeader = TRUE,
        "I am Body 2.",
        checkboxInput("check", label = "Hi, I am Checkbox 2")
       )
    )
  )
)

server <- function(input, output){ }
shinyApp(ui, server)
```



box

```
#2) Here is a very important box called tabBox. The tabBox is the most common function. This is simillar to the ta
bPanel of Shiny, but I think this tabBox is more convenient to use.

body <- dashboardBody(
   fluidPage(
      tabBox(
        title = "Box 1",
        id = "tab1", height = "250",
        tabPanel("You can type whatever you want to say in the box.",
                 textInput("text", label = "Hi, how are you? ")
        )
      )
   ),
  fluidPage(
    tabBox(
      title = "Box 2",
      id = "tab2", height = "250",
      tabPanel("Tab1", "It is sunny today."),
      tabPanel("Tab2", "I am fine.", br(), "Thank you.", br(), "And you?", br(), "(□˘□˘□)")
      )
  )
)

ui <- dashboardPage(
  dashboardHeader(title = "tabBoxes"),
    dashboardSidebar(),
    body
  )

server <- function(input, output){ }
shinyApp(ui, server)
```

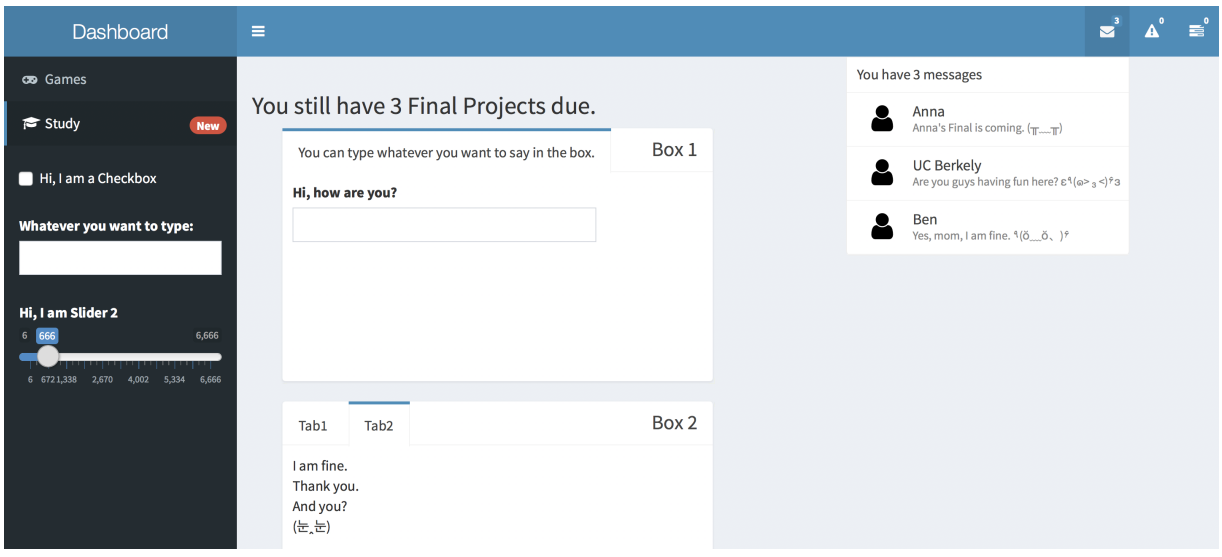


tabBox

6. The complete dashboard

After showing lots of functions and tools, I would like to show you a complete dashboard. Making our own dashboad is interesting.

```r
  header <- dashboardHeader(title = "Dashboard",
    dropdownMenu(type = "messages",
                 messageItem(
                   from = "Anna",
                   message = "Anna's Final is coming. (╥﹏╥) "
                 ),
                 messageItem(
                   from = "UC Berkely",
                   message = "Are you guys having fun here? ε٩(๑> ₃ <)۶з "
                 ),
                 messageItem(
                   from = "Ben",
                   message = "Yes, mom, I am fine. ٩(ŏ﹏ŏ、)۶ "
                 )),
    dropdownMenu(type = "notification"),
    dropdownMenu(type = "task")
  )

  sidebar <- dashboardSidebar(
    sidebarMenu(
      menuItem("Games", tabName = "games", icon = icon("gamepad")),
      menuItem("Study", tabName = "study", icon = icon("graduation-cap"),
               badgeLabel = "New", badgeColor = "red")
    ),
    checkboxInput("check", label = "Hi, I am a Checkbox"),
    textInput("text", label = "Whatever you want to type: "),
    sliderInput("slider", "Hi, I am Slider 2", min = 6, max = 6666, value = 666)
  )

  body <- dashboardBody(
      tabItems(
        tabItem(tabName = "study",
                h3("You still have 3 Final Projects due."),
                fluidPage(tabBox(
                  title = "Box 1", id = "tab1", height = "250",
                  tabPanel("You can type whatever you want to say in the box.",
                  textInput("text", label = "Hi, how are you? "))
                          )
                ),
                fluidPage(tabBox(
                  title = "Box 2", id = "tab2", height = "250",
                  tabPanel("Tab1", "It is sunny taday."),
                  tabPanel("Tab2", "I am fine.", br(), "Thank you.",
                           br(), "And you?", br(), "(눈_눈)")
                          )
                )
          ),
      tabItem(tabName = "games",
                h3("You don't have time to play games now.")
        )
      )
  )

ui <- dashboardPage(
    header,
    sidebar,
    body
  )

server <- function(input, output){ }
shinyApp(ui, server)
```

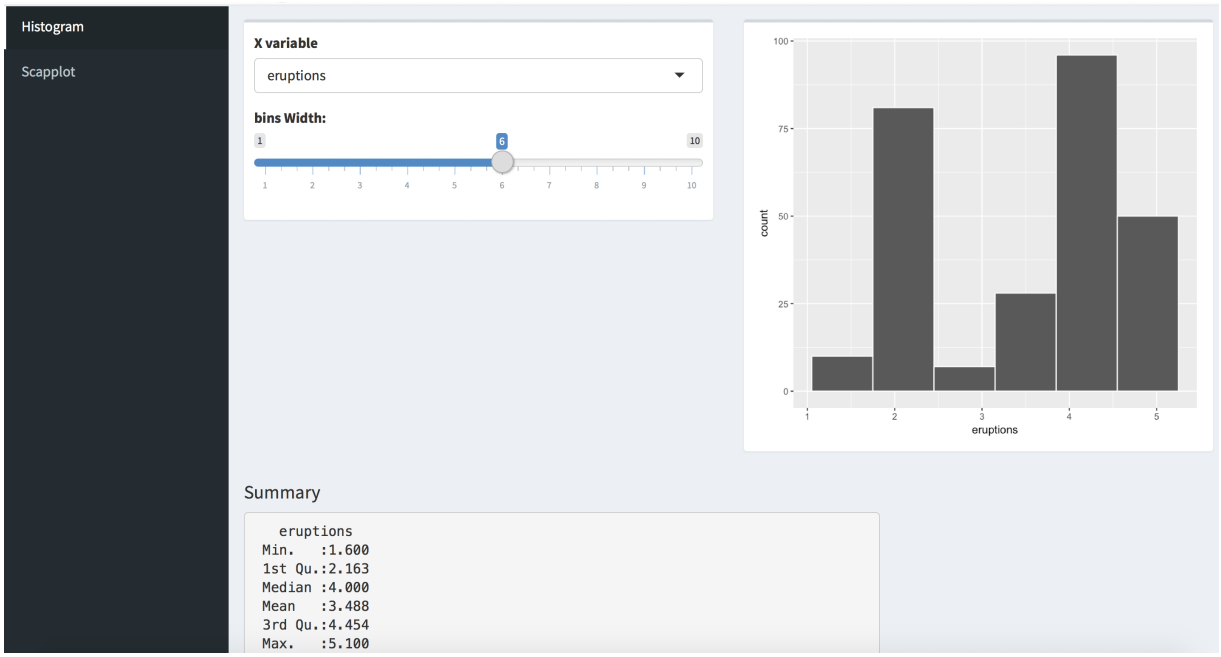


7. Application

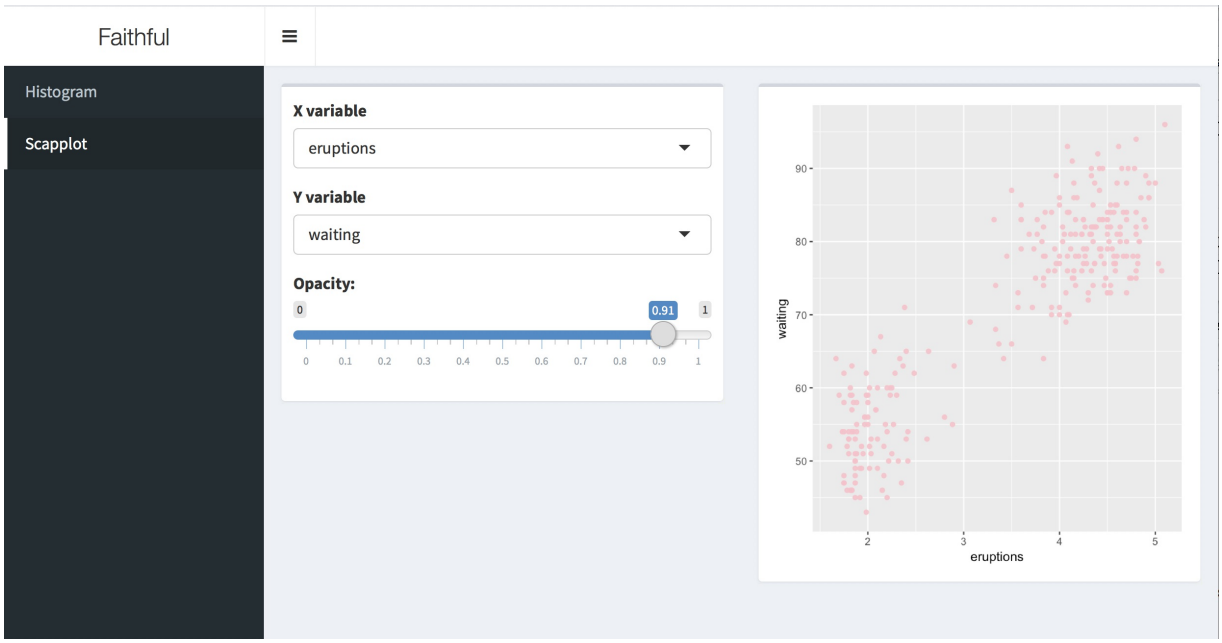
Right now, I would like to show you the real application by using 'shinydashboard'. I use the data called **faithful from RStudio database.

```
library(shiny)
library(shinydashboard)

ui <- dashboardPage(
  skin="black",

  #Title
  dashboardHeader(title = "Faithful",titleWidth = 222),

  #Sidebar
  dashboardSidebar(
    width = 221,
    sidebarMenu(
      menuItem("Histogram", tabName = "one"),
      menuItem("Scapplot", tabName = "two")
    )),

  #Content within the tabs
  dashboardBody(
    tabItems(
      tabItem(tabName = "one",
              fluidRow(
                box(
                 selectInput("x_variable1", "X variable", 'eruptions'),

                 sliderInput(inputId = "bins",
                             label = "bins Width:",
                             min = 1,
                             max = 10,
                             value = 6)
                ),
      box(plotOutput("plot1")),
      mainPanel(
        h4('Summary'),
        verbatimTextOutput('sum')
      )
              )
    ),

    tabItem(tabName = "two",
            fluidRow(
              box(
                selectInput("x_variable2", "X variable", 'eruptions'),
                selectInput("y_variable2", "Y variable", 'waiting'),

                sliderInput(inputId = "alpha1",
                            label = "Opacity:",
                            min = 0,
                            max = 1,
                            value = 0.5)
              ),
              box(plotOutput("plot2"))
            )
    )
  )
))


library(ggplot2)
mydata <- faithful
server <- function(input, output) {

  #one
  x_variable1 <- reactive({
    input$x_variable1
  })

  output$plot1 <- renderPlot({
    ggplot(data = mydata, aes_string(x = x_variable1())) +
      geom_histogram(bins = input$bins, color = "white")
  })

  summarydata <- reactive({
    temp <- subset(mydata,select = x_variable1())
    temp
  })

  output$sum <- renderPrint({
    summary(summarydata())
```

```
  })


  #two
  x_variable2 <- reactive({
    input$x_variable2
  })

  y_variable2 <- reactive({
    input$y_variable2
  })

  output$plot2 <- renderPlot({
      ggplot(data = mydata, aes_string(x = x_variable2(), y = y_variable2())) +
        geom_point(alpha = input$alpha1, color = "pink")
  })
}

shinyApp(ui, server)
```



1



2

8. take home message

The post is mainly about how to use shinydashboard package to create the dashboard. It is a very useful and esaier package. The most important thing is to remenber the structure of this package. There are three parts, header, sidebar and body. If you have a clear complete structure of what you want you do, you can save time. All the code in this post is computationally reproducible. Feel free to use them!

Okay~ It is time to build your own dashboard~ (*´∀`)~♥ Thanks for your reading.

Reference

1. http://shiny.rstudio.com
2. https://zhuanlan.zhihu.com/p/23596143
3. http://rstudio.github.io/shinydashboard/structure.html
4. http://fontawesome.io/icon/gamepad/
5. http://fontawesome.io/icon/graduation-cap/
6. https://home.gamer.com.tw/creationDetail.php?sn=3123042
7. https://www.google.com/search?q=funny+meme&client=safari&rls=en&tbm=isch&tbo=u&source=univ&sa=X&ved=0ahUKEwitubyt7OrXAhUrqFQKHfkFDpUQsAQIJg&biw=1440&bih=839
8. https://shiny.rstudio.com/gallery/
9. https://shiny.rstudio.com/gallery/kmeans-example.html