

R MARKDOWN: DASHBOARDS

Using shiny with flexdashboard

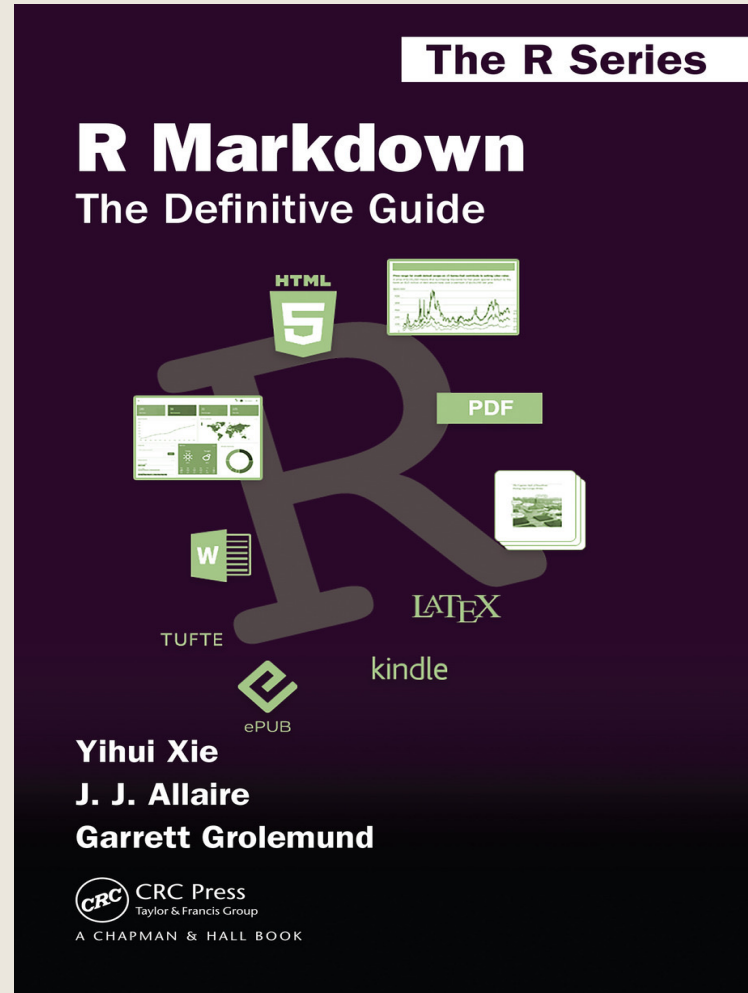
Geli Juani

June 16, 2021

Outline

- Using shiny with flexdashboard
- Input and output functions
- Sidebar
- Examples

Reference



[*R Markdown: The Definitive Guide*](#)

By the end of this session...

Make the following **interactive** dashboard

- Saudi Arabia Trade Charts Dashboard

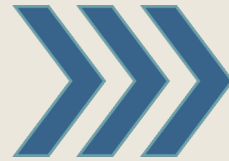
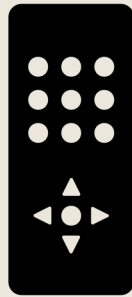
Using Shiny with Flexdashboard

- The flexdashboard package publishes figures, tables in a dashboard.
- The shiny package ([Chang et al. 2020](#)) builds interactive web apps powered by R.
 - `install.packages("flexdashboard")`
 - `install.packages("shiny")`
- Combining flexdashboard with shiny using R Markdown language
 - Viewers can change underlying parameters and see the results immediately
 - Done by adding `runtime: shiny` to the options declared at the top of the document (YAML).

```
---  
title: "Shiny Markdown Dashboard Output Types"  
output:  
  flexdashboard::flex_dashboard:  
    orientation: columns  
    runtime: shiny  
---
```

Using Shiny with Flexdashboard

User Input



Reactive Output



Input

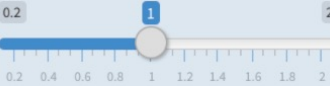
Old Faithful Eruptions

Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

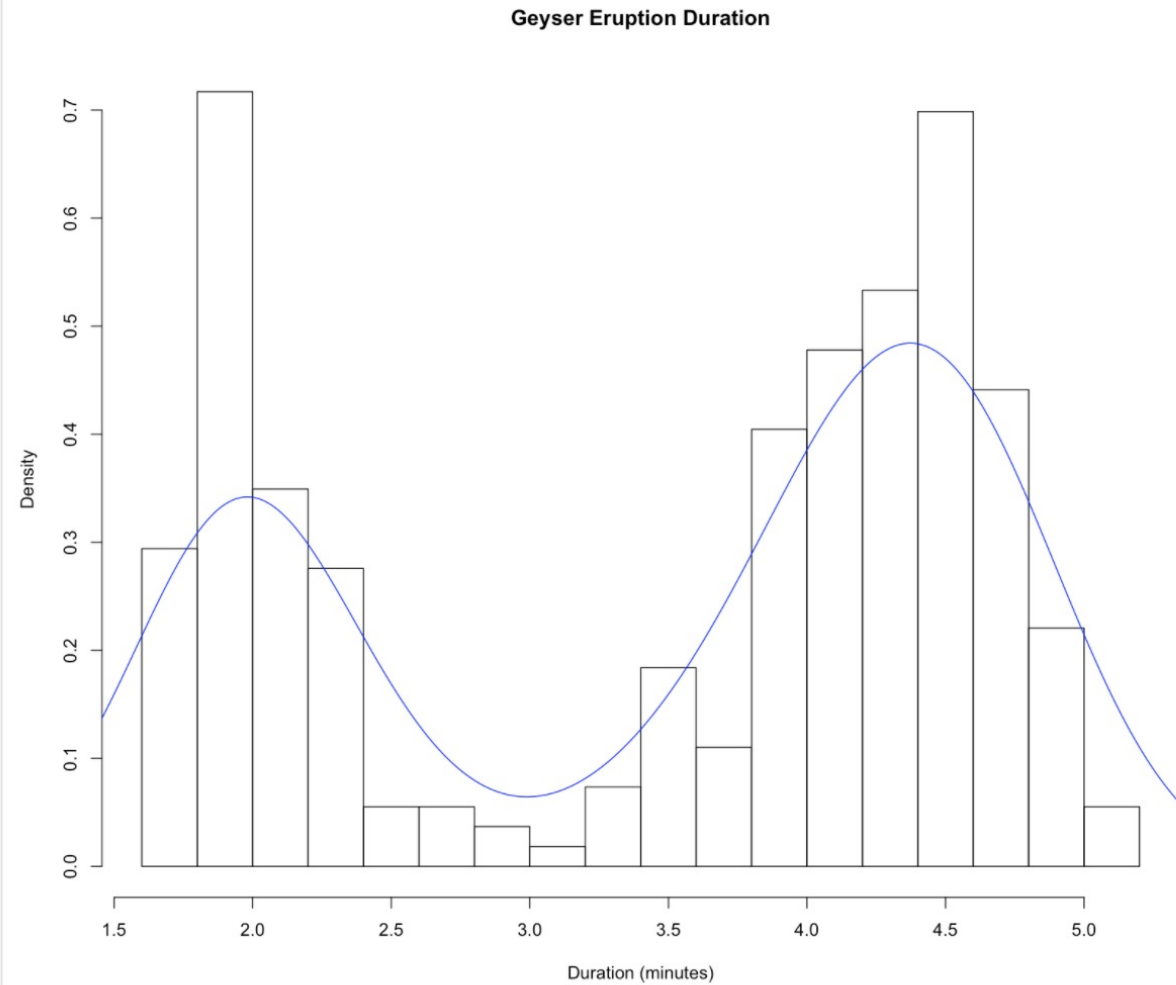
Number of bins:

20

Bandwidth adjustment:



Geyser Eruption Duration



Output

User Input

```
```{r}
selectInput(inputId="n_breaks", label = "Number of bins:",
 choices =c(10,20,35,50), selected=20)
sliderInput(inputId="bw_adjust", label = "Bandwidth adjustment:",
 min =0.2, max=2, value=1, step=0.2)
```
```

Reactive Output

```
```{r}
renderPlot({ hist(faithful$eruptions, probability = TRUE, breaks = as.numeric(input$n_breaks), xlab=
"Duration (minutes)", main = "Geyser Eruption Duration")

dens = density(faithful$eruptions, adjust = input$bw_adjust)
lines(dens, col = "blue")
})
```
```


Create user input controls

R Function

[selectInput](#)

[sliderInput](#)

[checkboxInput](#)

[radioButtons](#)

[dateInput](#)

[dateRangeInput](#)

[textInput](#)

[numericInput](#)

[passwordInput](#)

[fileInput](#)

Input Type

A dropdown box with choices to select from

A slider bar

A single check box

A set of radio buttons

A calendar to aid date selection

A pair of calendars for selecting a date range

A field to enter text

A field to enter numbers

A field to enter password

A file upload control wizard

Create reactive outputs

R Function

Output Type

[renderText](#)

Character vectors

[renderPrint](#)

R printed output

[renderTable](#)

Data frame, matrix, other table like structures

[renderPlot](#)

R graphics output

[renderValueBox](#)

A single value with a title and an icon

[renderGauge](#)

A numeric value on a meter

Sidebar

- You add an input sidebar to a flexdashboard by adding the `{.sidebar}` attribute
 - Sidebar always appears on the left
 - You can alter the default width of the sidebar using the `{data-width}` attribute
- Global sidebar
 - A single sidebar that applies across all dashboard pages
 - Define the sidebar using a level 1 markdown header (=====)

Sample dashboards

- Saudi Arabia Trade Charts Dashboard
 - *Using reactive function*

Deployment

- There are two ways to deploy a Shiny document
 1. Use a hosted service provided by Rstudio
 - [Shiny Server](#) and [RStudio Connect](#) can be used to publish Shiny documents
 2. Set up your own server

Take a step further

- Try out these [examples](#)
- Ready to develop your own shiny apps?
 - See [Mastering Shiny](#) guide by Hadley Wickham (2020)

R MARKDOWN: DASHBOARDS

Using shiny with flexdashboard

Geli Juani

June 16, 2021