

Outline

- R Shiny: Basics, How Apps Work, Getting Started, Sharing Apps
- Applications in Clinical Trials
- Some Helpful Resources
- Live Demo

R Shiny Basics

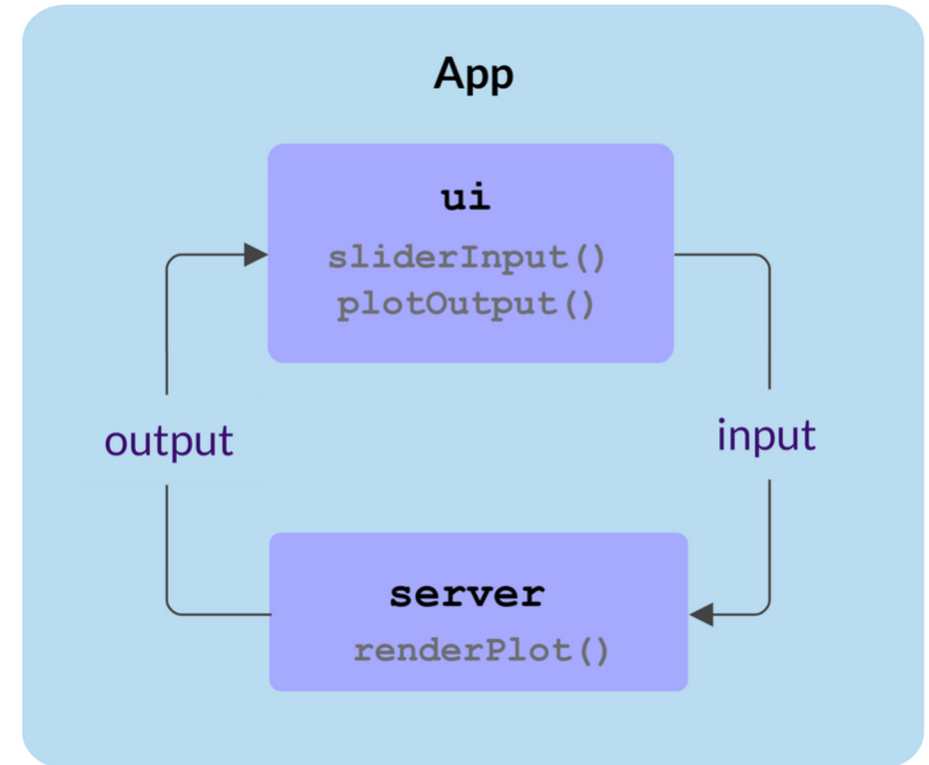
- Framework for creating web applications using R
- Interactivity!
 - Allows for users to interact with apps without any coding knowledge
 - Can help facilitate interdisciplinary collaboration
- Only knowledge of R is needed to build apps
 - Shiny takes care of the rest
 - No need to know HTML, CSS, or JavaScript
- High degree of flexibility
- Used in many disciplines, across academia and industry



How R Shiny Apps Work

Two main components:

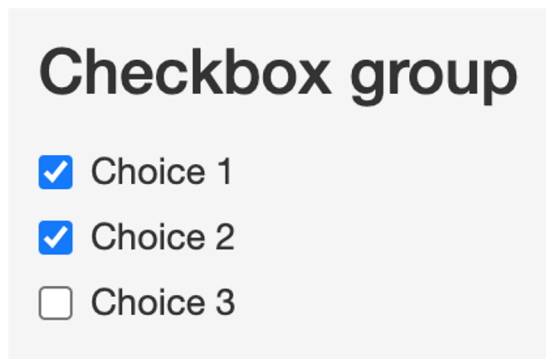
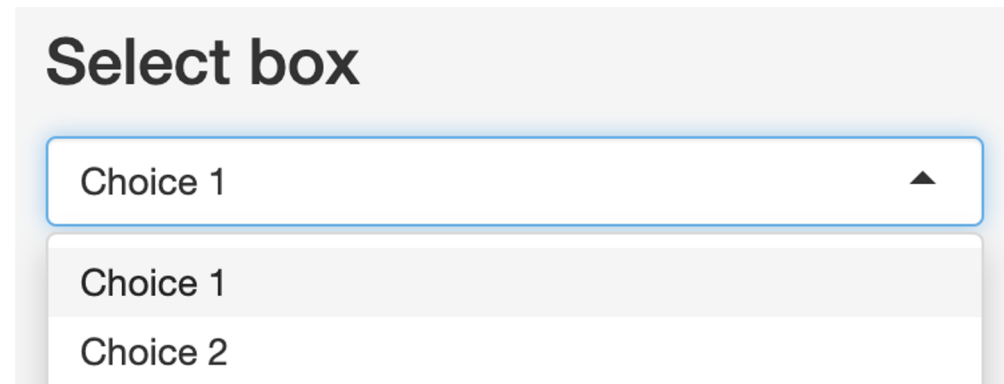
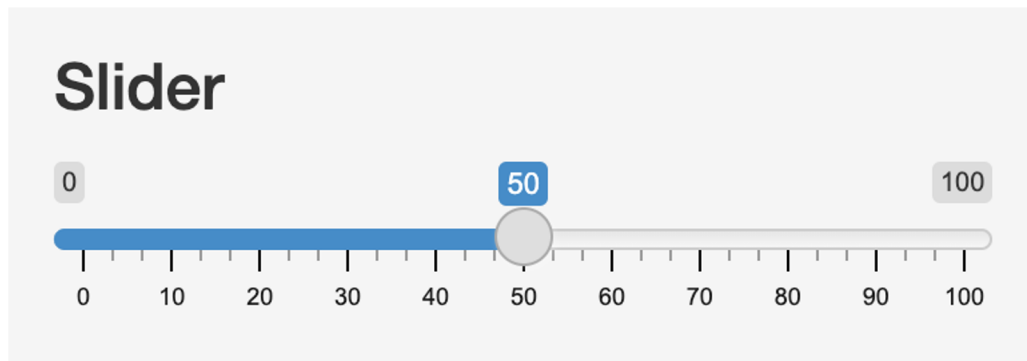
- UI (Front End Interface)
 - Obtain inputs via widgets
 - Display outputs
 - Control layout and appearance
- Server (Back End Logic)
 - Reacts to inputs and generates outputs



Source: <https://hosting.analythium.io/the-anatomy-of-a-shiny-application/>

UI: Obtain Inputs Via Widgets

- Users provide inputs by interacting with widgets
- Many different kinds of widgets:



*Plus file upload,
and much more...*

Server: Generate Outputs

Functions that react to inputs via widgets and generate outputs:

- `renderText()` – outputs text
- `renderTable()` – outputs table
- `renderPlot()` – outputs plot made using base R, `ggplot2`, ...
- `renderPlotly()` – output plot made using `plotly`

...

UI: Display Outputs

Functions that display outputs generated via the server code:

- `textOutput()` – displays text
- `tableOutput()` – displays table
- `plotOutput()` – displays plot made using base R, `ggplot2`, ...
- `plotlyOutput()` – displays plot made using `plotly`

...

Getting Started

Install the necessary components:

- R
- RStudio
- 'shiny' package

Two ways UI and Server can be structured in your code:

- Together in one .R file
- Two separate .R files (ui.R and server.R)

Take advantage of templates and previously created apps!

Structure Option 1 (Combined)

```
library(shiny)
ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)
server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
shinyApp(ui = ui, server = server)
```

Structure Option 2 (Separate)

```
# ui.R
fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)
```

```
# server.R
function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}
```

How to Share R Shiny Apps

1. Provide files

- Send raw files (.R, data, etc.) via email or similar means
- Recipient runs the app through RStudio
- Might not be feasible if collaborators do not have experience with R

2. Host on a server

- Free hosting on <https://www.shinyapps.io/> (paid plans offered as well)
 - Note: Be careful with sensitive data or intellectual property!
- Organizations can build their own servers (eg, AWS)

Applications in Clinical Trials

- *During the design phase*
 - Using simulation to select an optimized study design – particularly helpful for adaptive designs
- *As a trial is ongoing*
 - Ongoing monitoring of safety data
- *After a trial has completed*
 - Visualize and summarize final data

Some Helpful Resources

- Professor Jeff Goldsmith's P8105 Data Science I website: <https://www.p8105.com/shiny.html>
- RStudio Shiny App Gallery: <https://shiny.rstudio.com/gallery/>
- RStudio Shiny Widget Gallery: <https://shiny.rstudio.com/gallery/widget-gallery.html>
- RStudio Shiny "Cheat Sheet": <https://shiny.rstudio.com/images/shiny-cheatsheet.pdf>
- RStudio Shiny Tutorial: <https://shiny.rstudio.com/tutorial/>
- *Mastering Shiny* by Hadley Wickham: <https://mastering-shiny.org/>
- Google 😊

Live demo...