Applied Statistics for Data Scientists with R

Class 20: Introduction to R Shiny

Introduction

- It is a framework for developing web apps.
- First announced by RStudio in 2012.
- After the rebranding of RStudio PBC into Posit PBC, they launched shiny for Python in 2022.
- This is open source, and anyone who knows R can build shiny web app.
- You can freely host shiny apps in shinyapps.io

- https://shiny.posit.co/r/gallery/
- https://gallery.shinyapps.io/assistant

Requirements

- Rstudio IDE
- Package: shiny

- File -> New project -> New directory -> Shiny web application

- A basic shiny app usually contains these files:
  - app.R (or ui.R and server.R)
  - global.R
  - README
  - Description
  - /www
  - /`Other folders`

App Type Based on Files

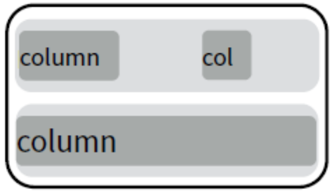
- Single file app
  - Name of the file should always be **app.R**
- Split file app
  - Name of the files should always be **ui.R** and **server.R**

Basic Components

- UI
  - ___Input() group of functions take inputs from the users.
  - ___Output() shows the outputs generated in the server.
- Server
  - render__() prepares output to show in UI.
  - UI elements get information from server from the **output$___** objects.

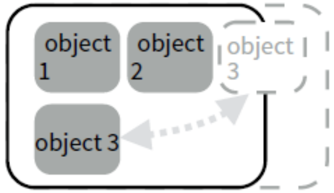
FluidPage layouts

fluidRow()



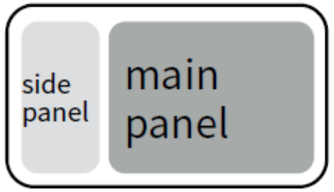
```
ui <- fluidPage(
 fluidRow(column(width = 4),
   column(width = 2,  offset = 3)),
 fluidRow(column(width = 12))
)
```

flowLayout()



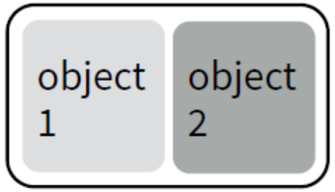
```
ui <- fluidPage(
 flowLayout( # object 1,
             # object 2,
             # object 3
 )
)
```

sidebarLayout()



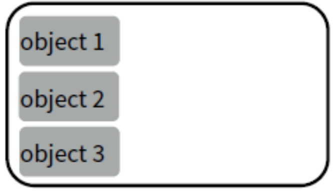
```
ui <- fluidPage(
  sidebarLayout(
    sidebarPanel(),
    mainPanel()
  )
)
```

splitLayout()



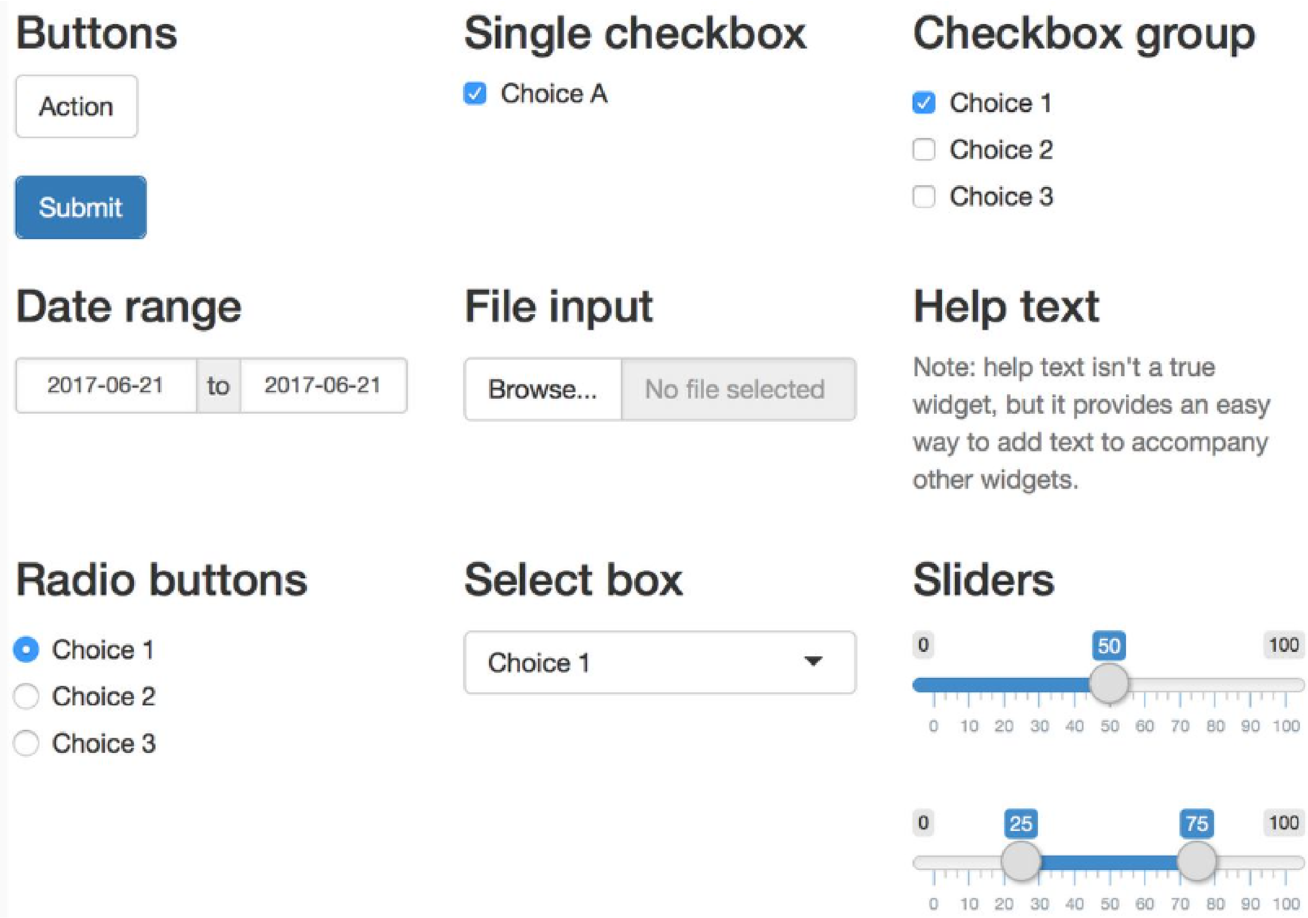
```
ui <- fluidPage(
 splitLayout( # object 1,
              # object 2
 )
)
```

verticalLayout()



```
ui <- fluidPage(
 verticalLayout( # object 1,
                 # object 2,
                 # object 3
 )
)
```

User Inputs



Buttons

Single checkbox

☑ Choice A

Checkbox group

☑ Choice 1
□ Choice 2
□ Choice 3

Date input

2014-01-01

Date range

2017-06-21 to 2017-06-21

File input

Browse... No file selected

Help text

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

Numeric input

1

Radio buttons

● Choice 1
○ Choice 2
○ Choice 3

Select box

Choice 1

Sliders

Text input

Enter text...

https://shiny.posit.co/r/gallery/widgets/widget-gallery/

User Inputs

- numericInput: field to enter numbers
- selectInput: box with choices to select from
- sliderInput: slider bar
- submitButton: submit button
- actionButton: action Button
- checkboxInput: single check box
- checkboxGroupInput: group of check boxes
- dateInput: calendar to aid date selection
- dateRangeInput: pair of calendars for selecting a date range
- fileInput: file upload control wizard
- helpText: help text that can be added to an input form
- radioButtons: set of radio buttons
- textInput: field to enter text

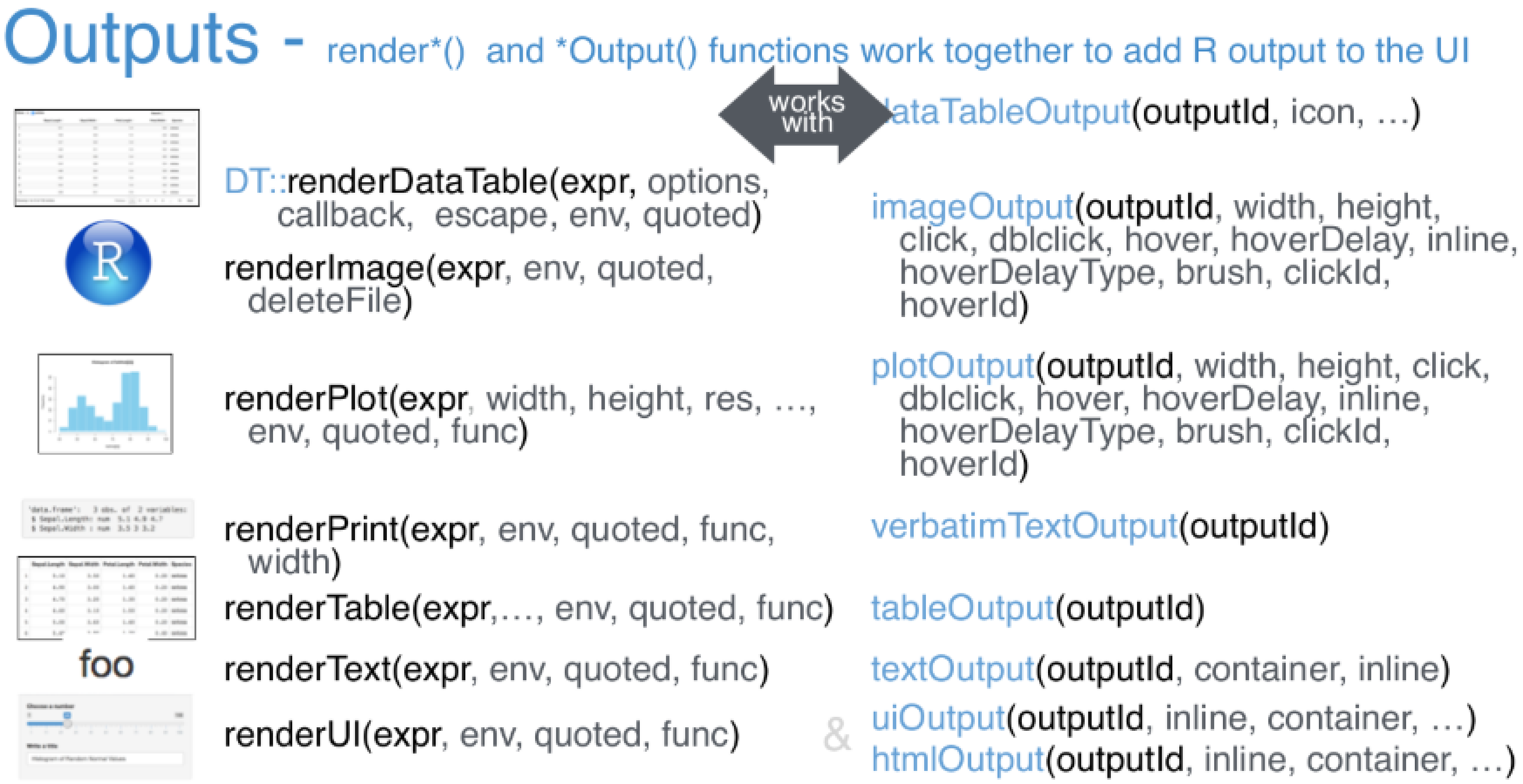
User Outputs

- dataTableOutput: DataTable
- tableOutput: table
- plotOutput: plot
- htmlOutput: raw HTML
- imageOutput: image
- textOutput: text
- uiOutput: raw HTML
- verbatimTextOutput: text

Render in Servers

- renderDataTable: DataTable
- renderImage: images (saved as a link to a source file)
- renderPlot: plots
- renderPrint: any printed output
- renderTable: data frame, matrix, other table like structures
- renderText: character strings
- renderUI: a Shiny tag object or HTML

Render in Servers


works with
foo

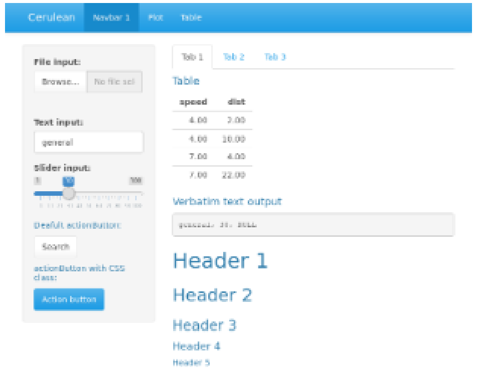
Outputs - render*() and *Output() functions work together to add R output to the UI

DT::renderDataTable(expr, options, callback, escape, env, quoted)		dataTableOutput(outputId, icon, …)
renderImage(expr, env, quoted, deleteFile)		imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)
renderPlot(expr, width, height, res, …, env, quoted, func)		plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, inline, hoverDelayType, brush, clickId, hoverId)
renderPrint(expr, env, quoted, func, width)		verbatimTextOutput(outputId)
renderTable(expr,…, env, quoted, func)		tableOutput(outputId)
renderText(expr, env, quoted, func)		textOutput(outputId, container, inline)

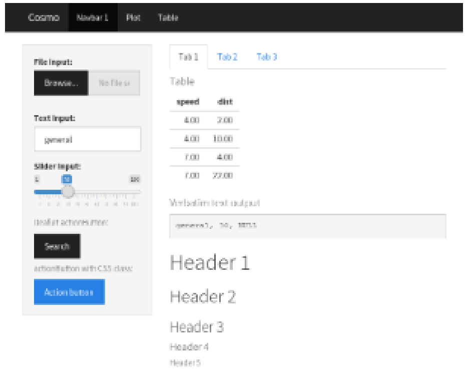
renderUI(expr, env, quoted, func)	&	uiOutput(outputId, inline, container, …)
		htmlOutput(outputId, inline, container, …)

Shiny App Themes

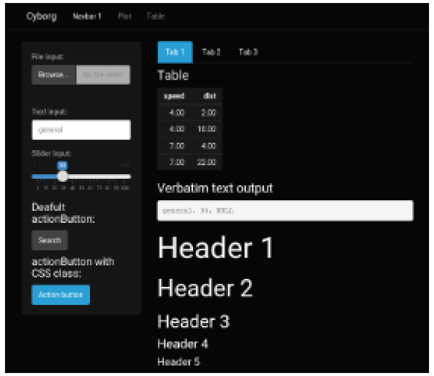
- https://rstudio.github.io/shinythemes/



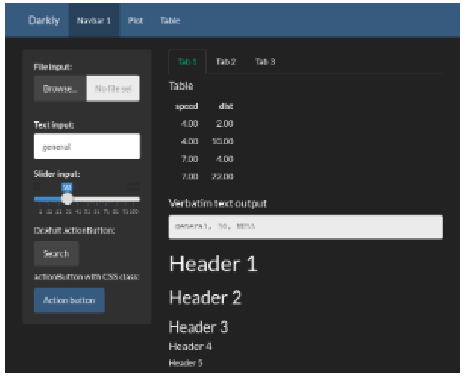
speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00



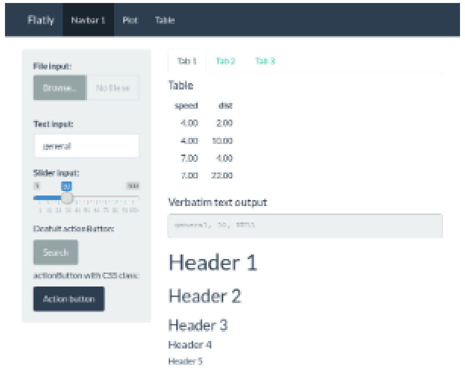
speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00



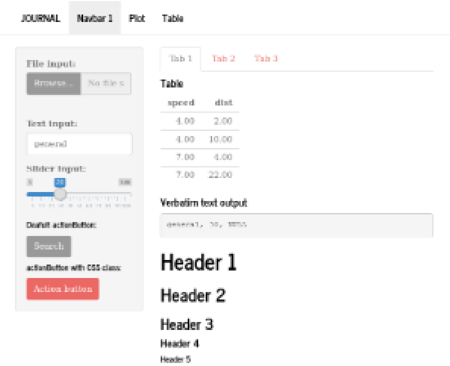
speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00



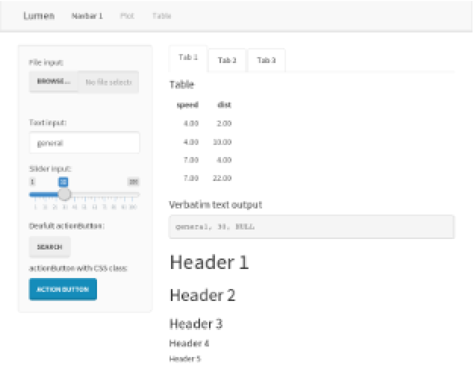
speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00



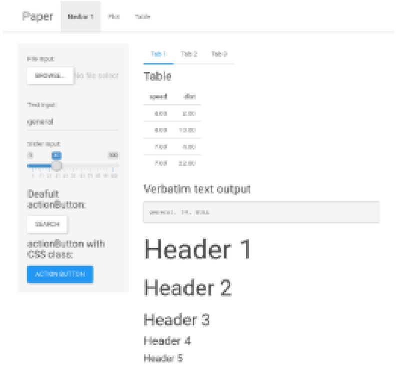
speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00



speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00

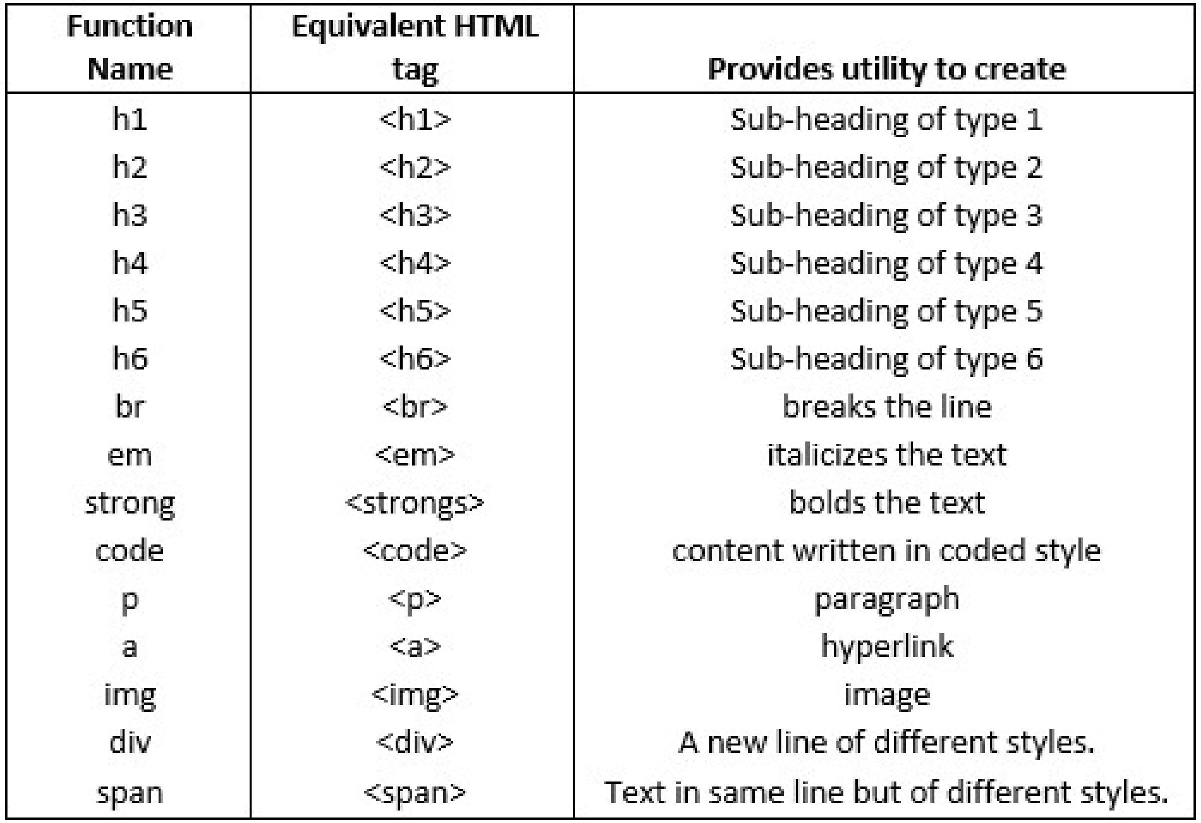


speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00



speed	dist
4.00	2.00
4.00	10.00
7.00	4.00
7.00	22.00

Useful HTML tags as functions



Function Name	Equivalent HTML tag	Provides utility to create
h1	<h1>	Sub-heading of type 1
h2	<h2>	Sub-heading of type 2
h3	<h3>	Sub-heading of type 3
h4	<h4>	Sub-heading of type 4
h5	<h5>	Sub-heading of type 5
h6	<h6>	Sub-heading of type 6
br	<br>	breaks the line
em	<em>	italicizes the text
strong	<strongs>	bolds the text
code	<code>	content written in coded style
p	<p>	paragraph
a	<a>	hyperlink
img	<img>	image
div	<div>	A new line of different styles.
span	<span>	Text in same line but of different styles.