Control widgets and reactivity

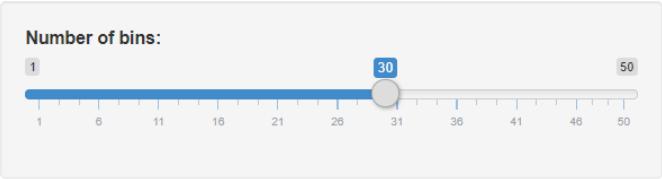
- Variety of control widgets
- Review reactivity
- Change default app.

  Add in an input controller for the colour of the histogram.

  Display text for the color of the histogram

Control widgets

The slider input is a control widget.




```
sliderInput("bins",
            "Number of bins:",
            min = 1,
            max = 50,
            value = 30)
```

app.R (reactivity)

Change the slider input value

```
# user interface (what you see when you run the app)
ui <- fluidpage(

 sliderInput("bins",  →  input$bins
             "Number of bins:",
             min = 1,
             max = 50,
             value = 20)
)

# calculations and plots (the usual R code)
server <- function(input, output) {

 bins <- seq(min(x), max(x), length.out = input$bins + 1)

}
```

sliderInput is a reactive input function. If the slider is moved then the input$bins value is updated.

There's a whole lot of reactive input functions with names:

XXXXXInput

app.R (reactivity)

An updated plot to show in the user interface

```
# user interface (what you see when you run the app)
ui <- fluidpage(

 mainPanel(
        plotOutput("distPlot")
      )

)

# calculations and plots (the usual R code)
server <- function(input, output) {

 output$distPlot <- renderPlot({

      # draw the histogram with the specified number of bins
      hist(x, breaks = bins, col = 'blue', border = 'white')
   })

}
```

renderPlot is a reactive function. If the plot inside the function changes then the user interface display is updated.

There's a whole lot of reactive output functions with names:

renderXXXXX

Control widgets: what they look like

Basic widgets

Buttons

Action

Submit

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

0 50 100
0 10 20 30 40 50 60 70 80 90 100

0 25 75 100
0 10 20 30 40 50 60 70 80 90 100

Text input

Enter text...

Control widgets: what they look like

Basic widgets

Buttons

Action

Submit

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

0 50 100

0 10 20 30 40 50 60 70 80 90 100

0 25 75 100

0 10 20 30 40 50 60 70 80 90 100

Text input

Enter text...

Control widgets: associated functions

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

(used in default app)

Control widgets: interactive gallery

Exercise. Use the **Select box** widget (selectInput function) and make an input control widget for the histogram colour: red, blue, green (make a new app for this)

Extra user interface code

```
selectInput(inputId = "hist_colour",
            label   = "Color of histogram: ",
            choices = c("Red" = "red",
                        "Blue" = "blue",
                        "Green" = "green")),
```

Change to server code

```
hist(x, breaks = bins, col = input$"hist_colour", border = 'white')
```

Control widgets: interactive gallery

Have a play at this link

https://shiny.rstudio.com/gallery/widget-gallery.html

Exercise. Use the **Select box** widget (selectInput function) and make an input control widget for the histogram colour: red, blue, green

render functions for server

Render functions are reactive, and if the input changes (e.g. a slider value), then the output from then changes (e.g. there's a new histogram)

render function	creates
`renderDataTable`	DataTable
`renderImage`	images (saved as a link to a source file)
`renderPlot`	plots (used in default app)
`renderPrint`	any printed output
`renderTable`	data frame, matrix, other table like structures
`renderText`	character strings
`renderUI`	a Shiny tag object or HTML

Output functions for user interface

Output function	Creates
`dataTableOutput`	DataTable
`htmlOutput`	raw HTML
`imageOutput`	image
`plotOutput`	plot
`tableOutput`	table
`textOutput`	text
`uiOutput`	raw HTML
`verbatimTextOutput`	text

(used in default app)

render and Output: exercise

Exercise. Put some text above the histogram giving the colour of the histogram

e.g. The selected colour is red

render and Output: exercise

Exercise. Put some text above the histogram giving the colour of the histogram

e.g. The selected colour is red

New user interface code

```
mainPanel(
        textOutput("selected_colour"),
        plotOutput("distPlot")
      )
```

New server code

```
 output$selected_colour <- renderText({
    paste("The selected colour is", input$"hist_colour")
   })
```