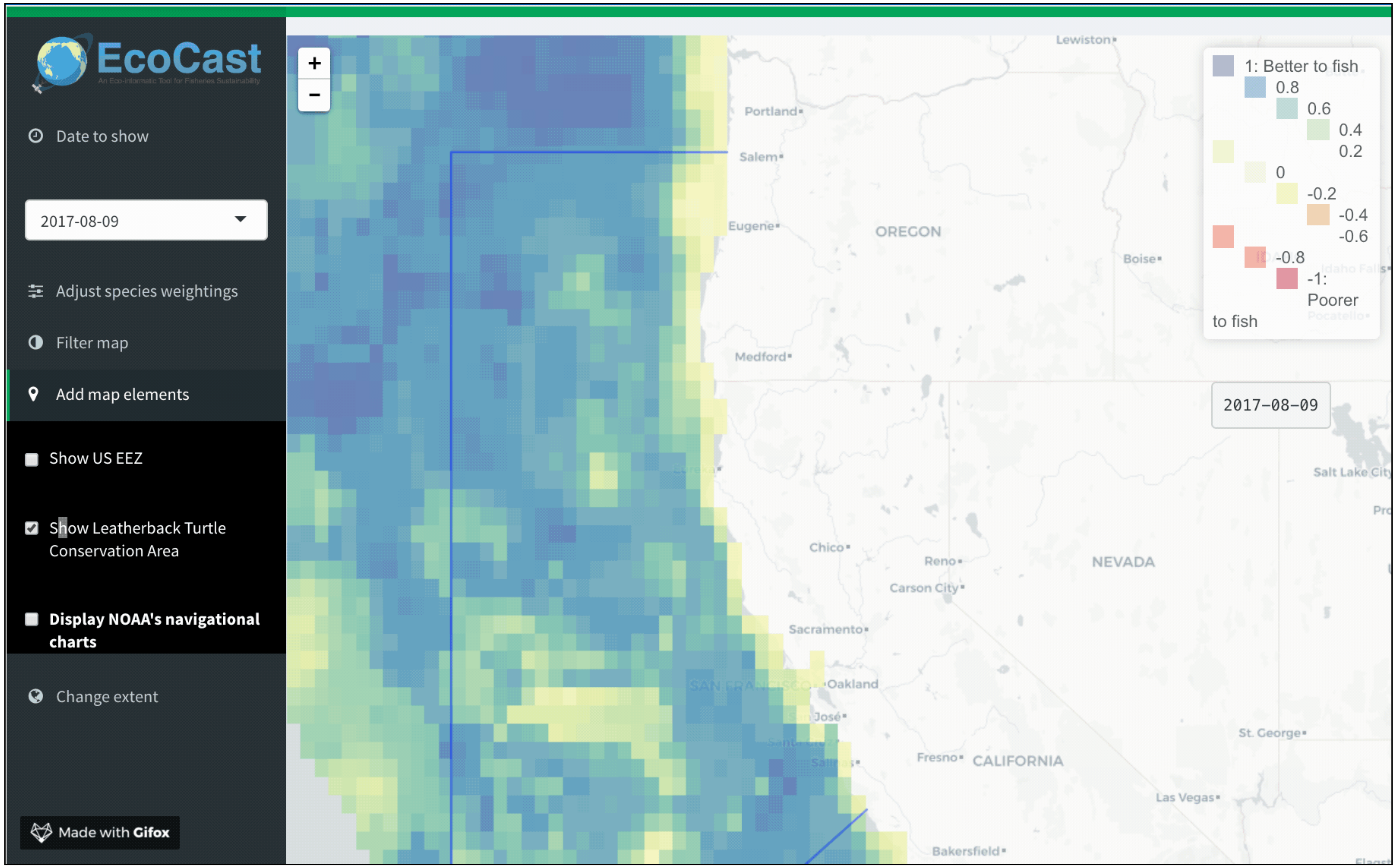
Intro to Shiny

An RLadies workshop
Megsie Siple
February 17, 2019

Why Shiny?

- Making tools for people
- Communicating science
- Teaching

Fisheries tools that use Shiny

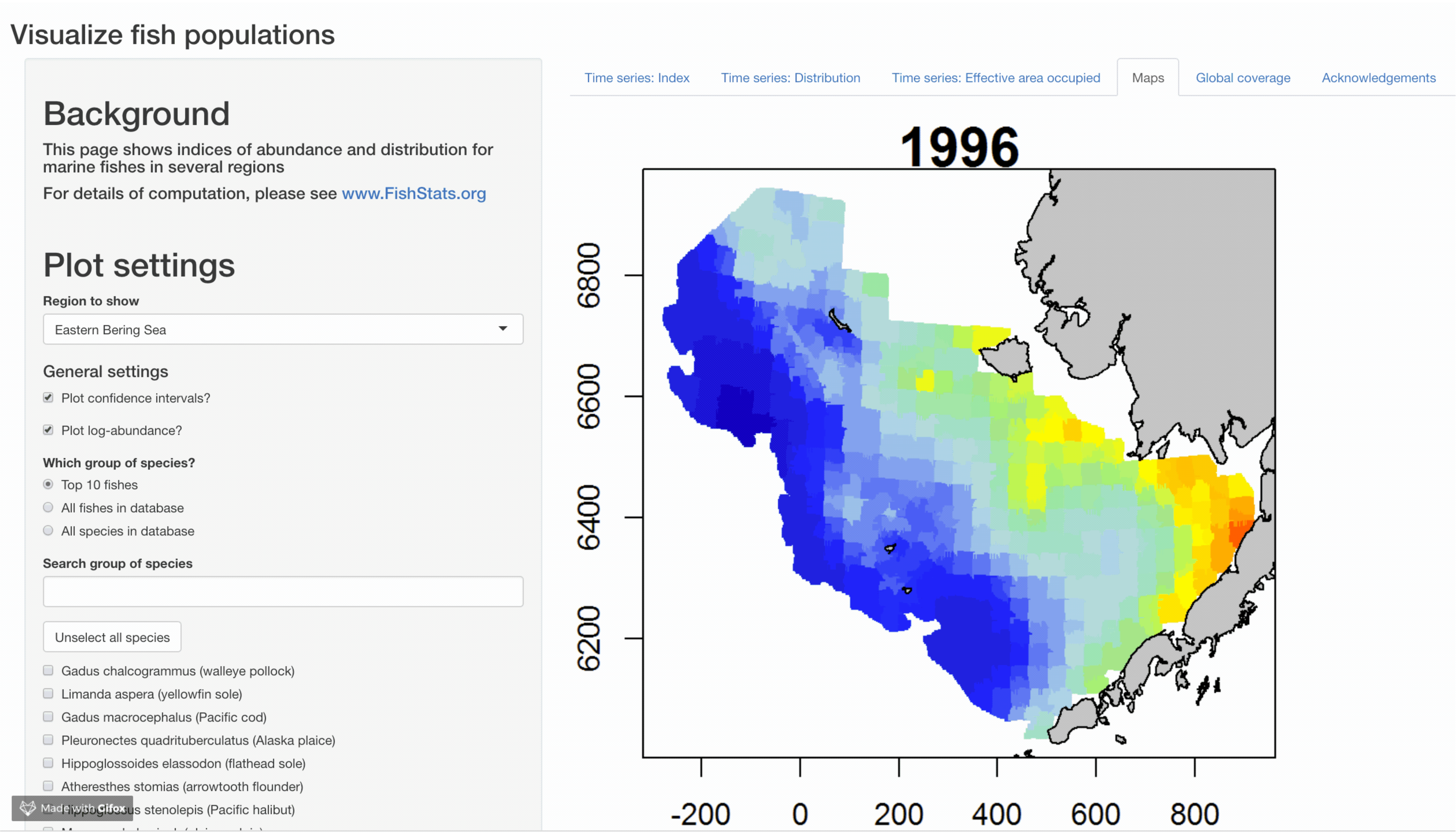


Date to show: 2017-08-09

☐ Show US EEZ

☑ Show Leatherback Turtle Conservation Area

☐ **Display NOAA's navigational charts**



Visualize fish populations

Background

This page shows indices of abundance and distribution for marine fishes in several regions

For details of computation, please see www.FishStats.org

Plot settings

Region to show

Eastern Bering Sea

General settings

☑ Plot confidence intervals?

☑ Plot log-abundance?

Which group of species?

- ● Top 10 fishes
- ○ All fishes in database
- ○ All species in database

Search group of species

- □ Gadus chalcogrammus (walleye pollock)
- □ Limanda aspera (yellowfin sole)
- □ Gadus macrocephalus (Pacific cod)
- □ Pleuronectes quadrituberculatus (Alaska plaice)
- □ Hippoglossoides elassodon (flathead sole)
- □ Atheresthes stomias (arrowtooth flounder)
- □ Hippoglossus stenolepis (Pacific halibut)

Goals for today!

- Learn the nuts and bolts of Shiny
- Learn how Shiny can be used for communicating data and/or results
- Learn the basics so you can incorporate it in your work
- Have fun and add to your programming toolbox

A workshop outline

10:00 - 11:00: The elements of Shiny

11:00 - 11:15: *Coffeeeeee* ☕

11:15 - 12:15: Working with different data types (time series and spatial data)

12:15 - 13:00: *Lunch*

13:00 - 14:00: Bringing data in and out of Shiny

14:00 - 14:15: *More coffee* ☕

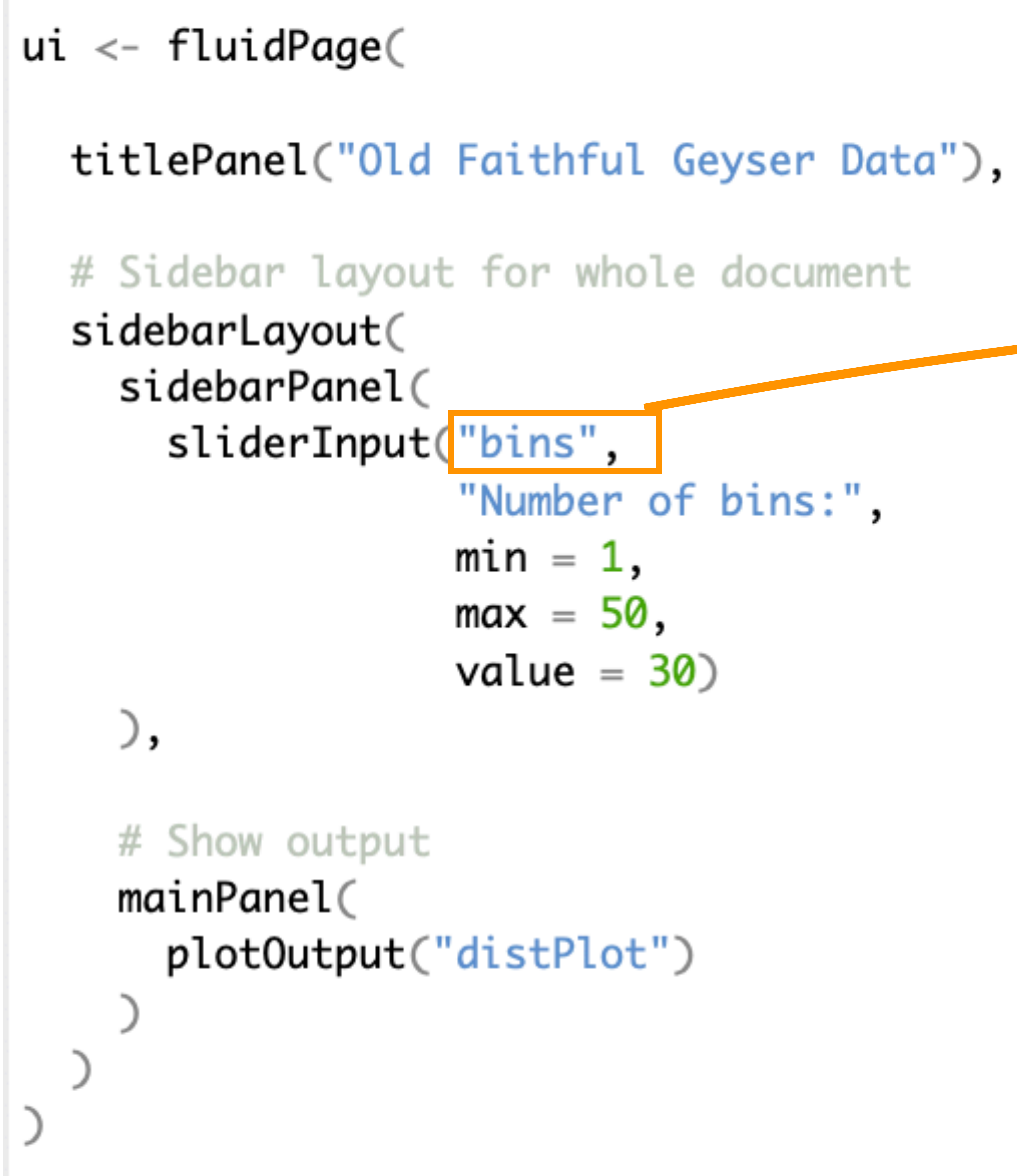
14:15 - 15:00: Sharing/publishing

How it works

- user interface & server logic
- ui shows inputs and outputs
- server performs all the operations
- super flexible! you can use CSS themes, html widgets, and JavaScript actions. It is awesome.

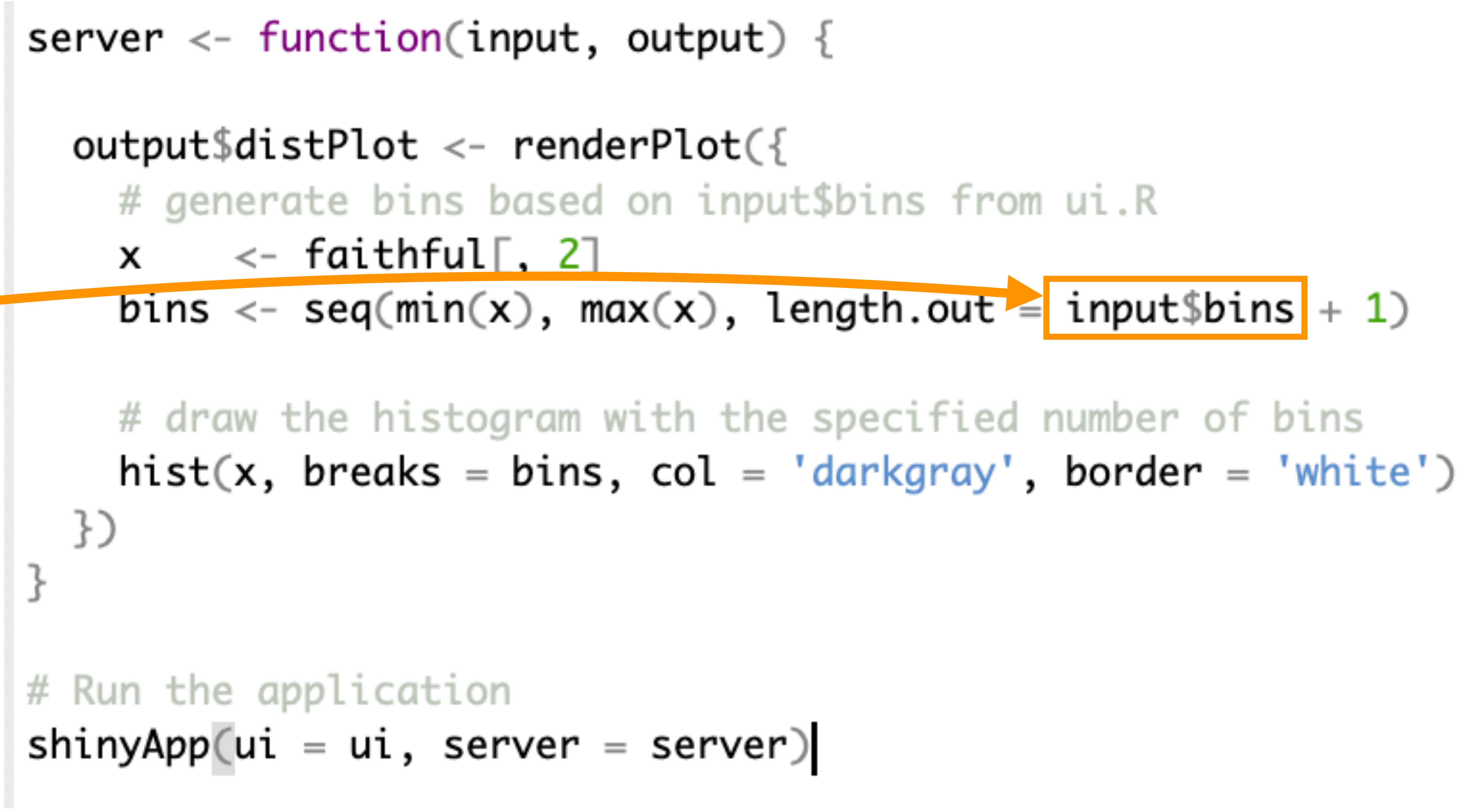
How it works

User interface



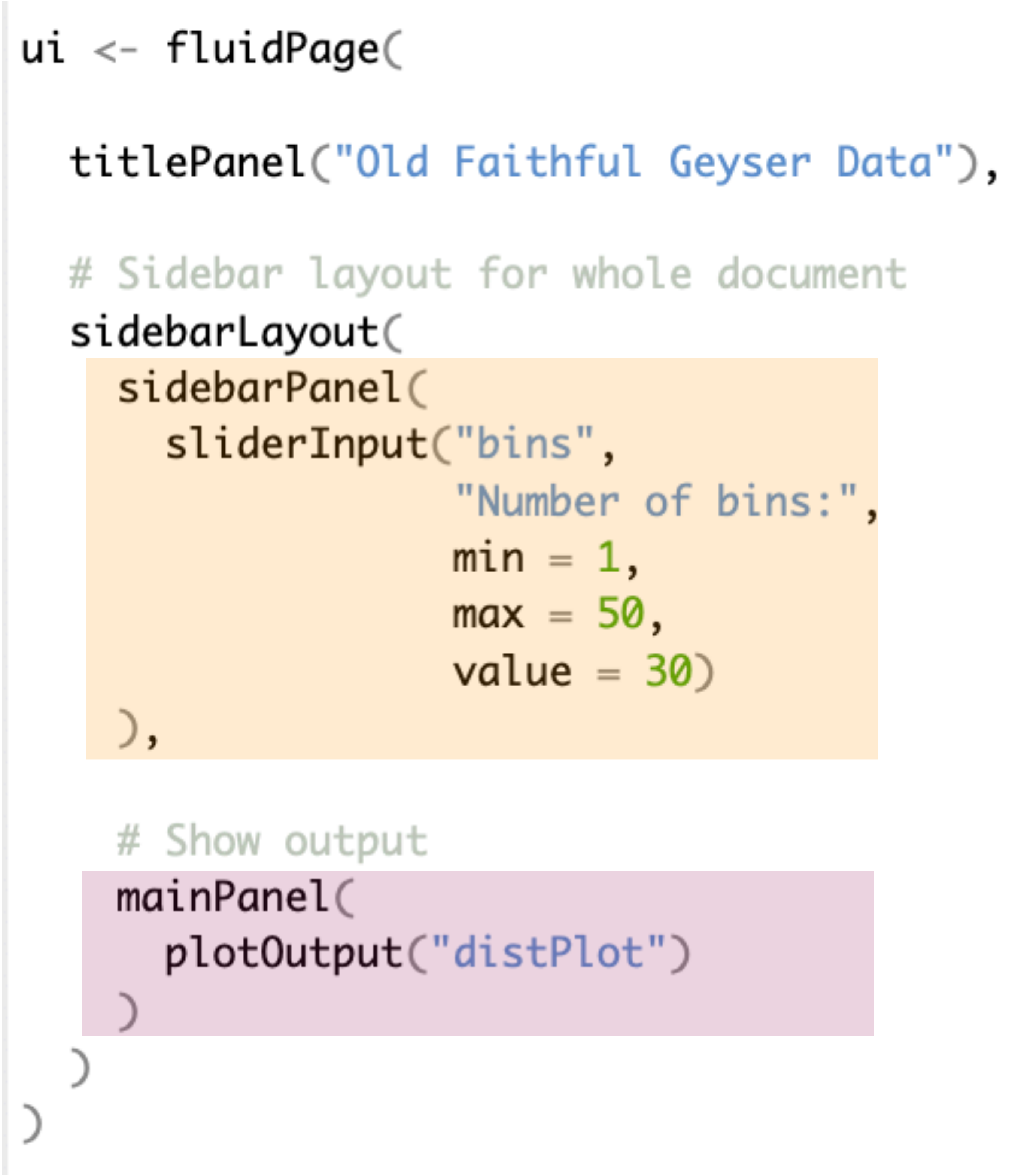
```
ui <- fluidPage(

  titlePanel("Old Faithful Geyser Data"),

  # Sidebar layout for whole document
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
    ),

    # Show output
    mainPanel(
      plotOutput("distPlot")
    )
  )
)
```

Server logic

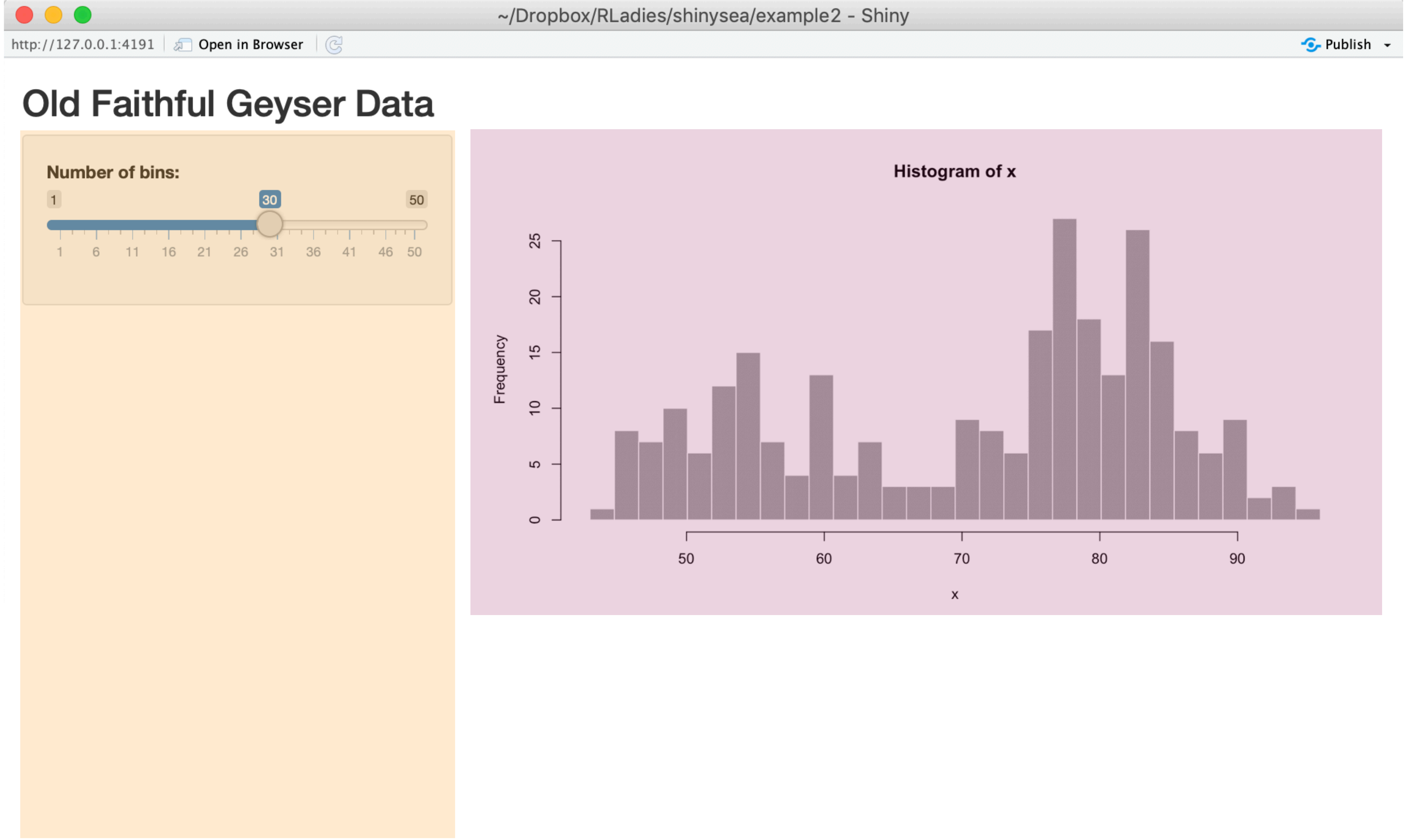


```
server <- function(input, output) {

  output$distPlot <- renderPlot({
    # generate bins based on input$bins from ui.R
    x    <- faithful[, 2]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

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

# Run the application
shinyApp(ui = ui, server = server)
```

User interface



```
ui <- fluidPage(

  titlePanel("Old Faithful Geyser Data"),

  # Sidebar layout for whole document
  sidebarLayout(
    sidebarPanel(
      sliderInput("bins",
                  "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)
    ),

    # Show output
    mainPanel(
      plotOutput("distPlot")
    )
  )
)
```



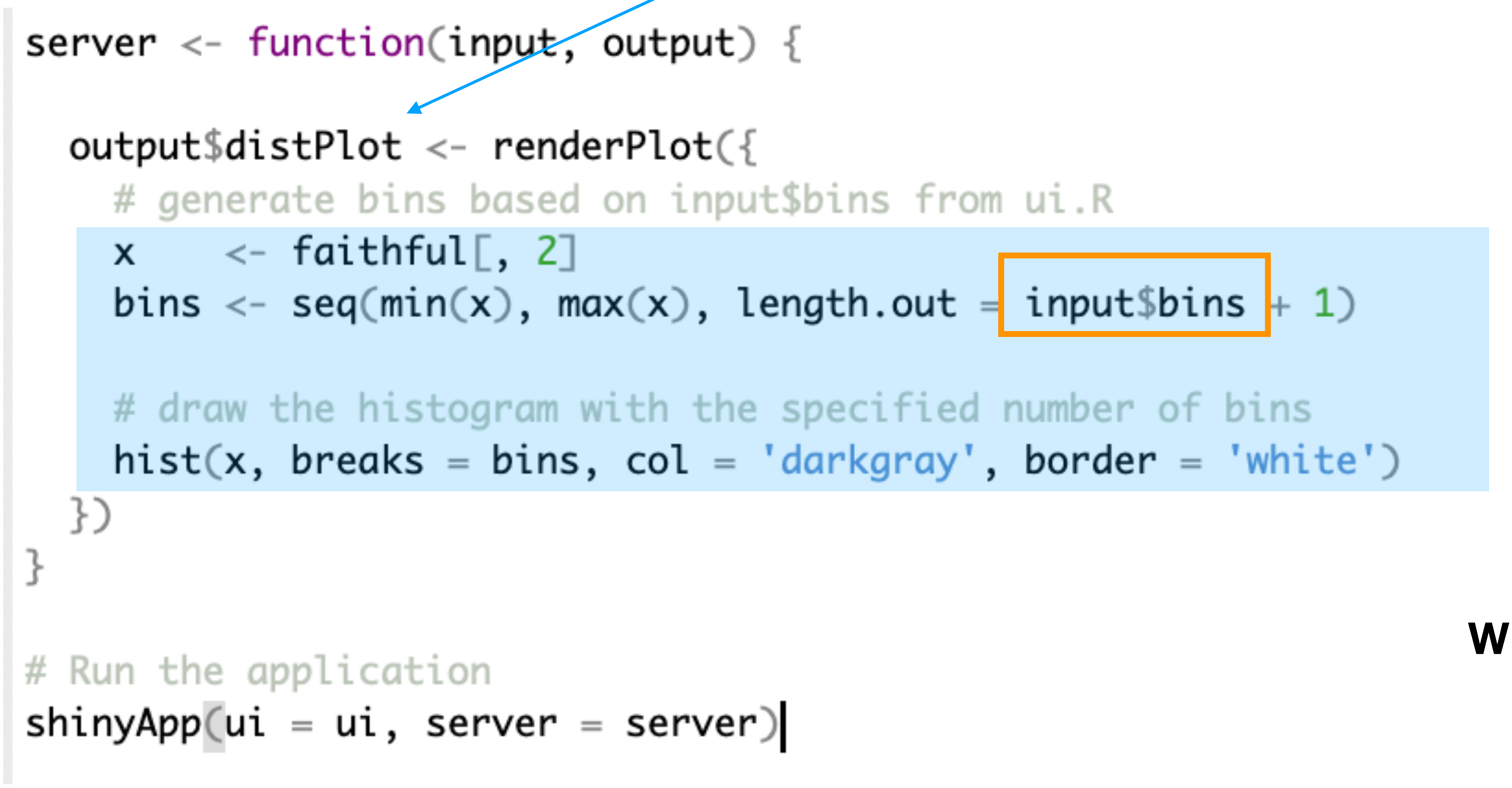
~/Dropbox/RLadies/shinysea/example2 - Shiny

http://127.0.0.1:4191 Open in Browser Publish

Old Faithful Geyser Data

Number of bins:

Server logic

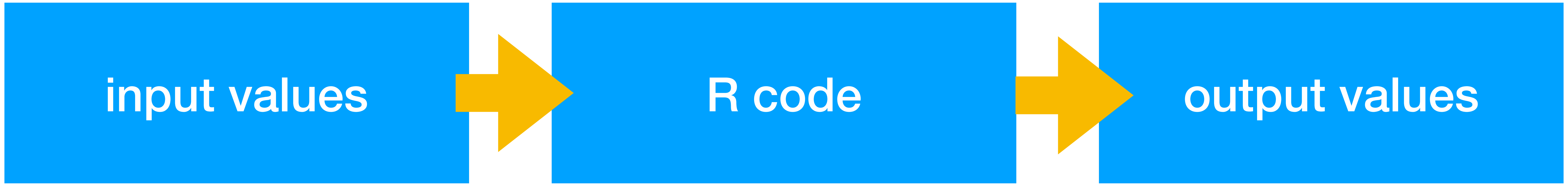


Output object

```
server <- function(input, output) {

  output$distPlot <- renderPlot({
    # generate bins based on input$bins from ui.R
    x    <- faithful[, 2]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

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

# Run the application
shinyApp(ui = ui, server = server)
```

Within each object, everything looks like plain old R
Except the pieces that connect to the UI

Reactivity

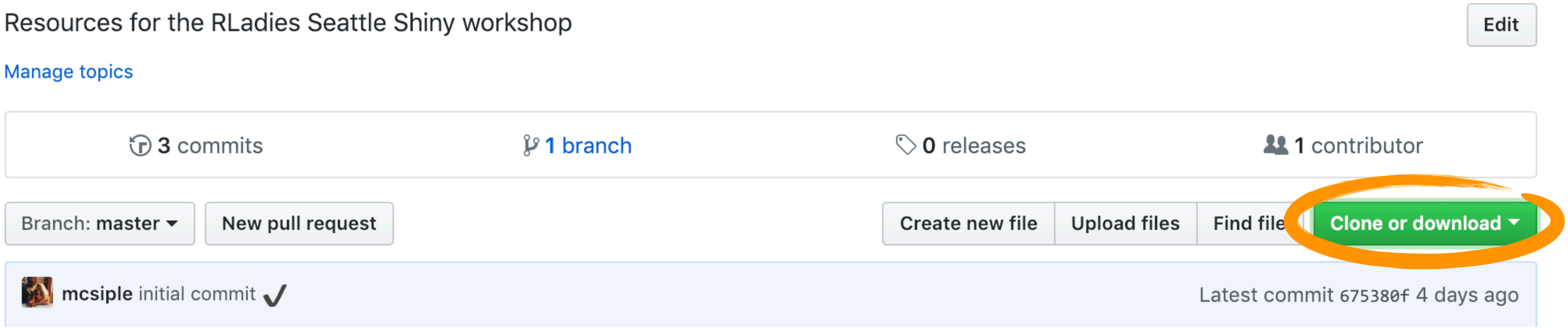


- Reactive programming starts with reactive values and executes reactive expressions

- So every time the reactive values change (i.e., when a user changes a value in the interface) the reactive expressions are immediately executed again (more on this later)

Download workshop files

1. Go to github.com/mcsiple/shinysea
2. Clone or download the whole repo
3. Now you have all the files you need!



Resources for the RLadies Seattle Shiny workshop

mcsiple initial commit ✓

Latest commit 675380f 4 days ago

Example 1: Orange trees

🍊