Shiny apps

- creates a UI
- web based
- uses R (and HTML/CSS)

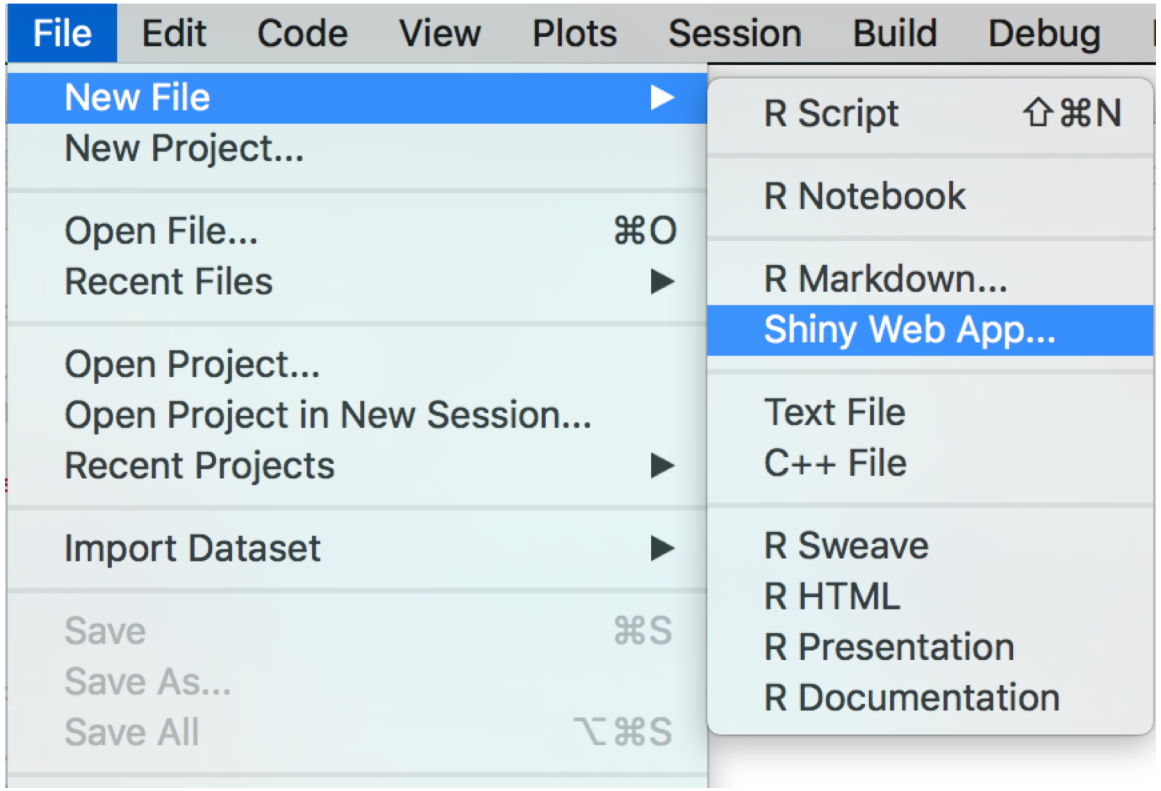
Examples

- BoxPlotR
- PlotsOfData
- Plasmid database

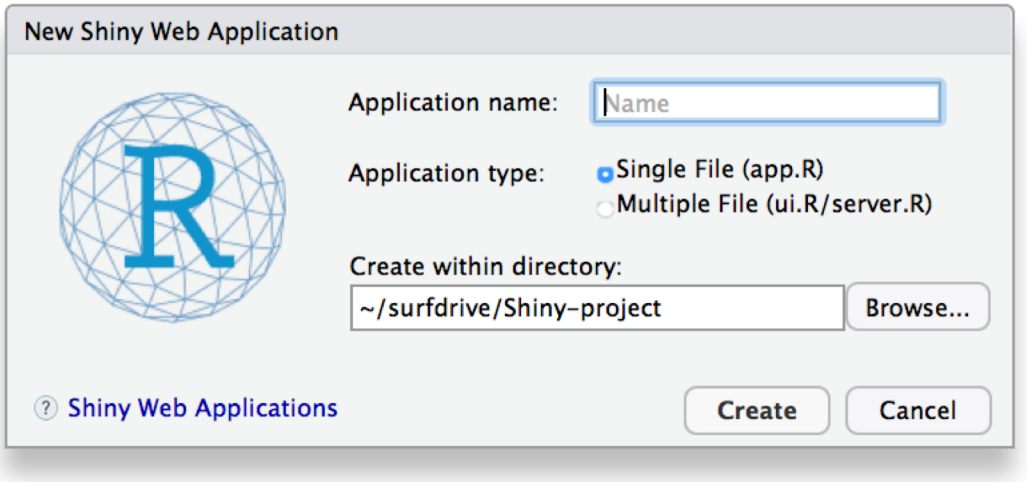
Why?

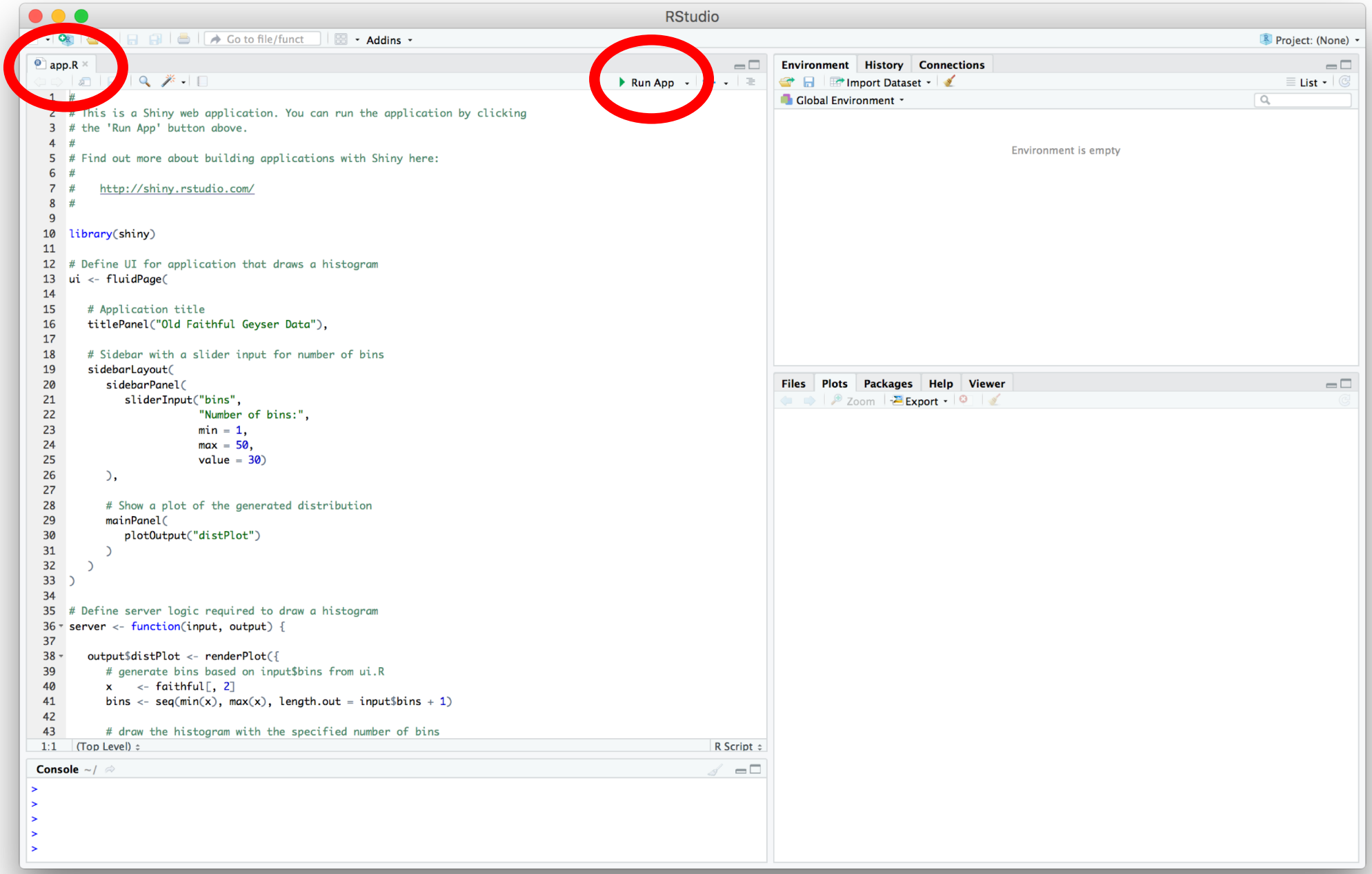
- Online (and offline) access
- Open source
- User friendly

First steps - RStudio



First steps - RStudio




RStudio
Go to file/funct
Addins
Project: (None)
app.R
Run App
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Zoom
Export
Console ~/
(Top Level)
R Script
1:1
#
This is a Shiny web application. You can run the application by clicking
the 'Run App' button above.
#
Find out more about building applications with Shiny here:
#
http://shiny.rstudio.com/
#
library(shiny)
# Define UI for application that draws a histogram
ui <- fluidPage(
   # Application title
   titlePanel("Old Faithful Geyser Data"),
   # Sidebar with a slider input for number of bins
   sidebarLayout(
      sidebarPanel(
         sliderInput("bins",
                     "Number of bins:",
                     min = 1,
                     max = 50,
                     value = 30)
      ),
      # Show a plot of the generated distribution
      mainPanel(
         plotOutput("distPlot")
)
)
)
# Define server logic required to draw a histogram
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

```r
#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)

# Define UI for application that draws a histogram
ui <- fluidPage(

   # Application title
   titlePanel("Old Faithful Geyser Data"),

   # Sidebar with a slider input for number of bins
   sidebarLayout(
      sidebarPanel(
         sliderInput("bins",
                     "Number of bins:",
                     min = 1,
                     max = 50,
                     value = 30)
      ),

      # Show a plot of the generated distribution
      mainPanel(
         plotOutput("distPlot")
      )
   )
)

# Define server logic required to draw a histogram
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)

```

```
20        sidebarPanel(
21           sliderInput("bins"
22                       "Number of bins:",
23                       min = 1,
24                       max = 50,
25                       value = 30)
```

```
39        # generate bins based on input$bins from ui.R
40        x    <- faithful[, 2]
41        bins <- seq(min(x), max(x), length.out = input$bins + 1)
42
43        # draw the histogram with the specified number of bins
44        hist(x, breaks = bins, col = 'darkgray', border = 'white')
```

```
30        plotOutput("distPlot")
```

```
38    output$distPlot <- renderPlot({
```

Building an app: step-by-step

https://github.com/JoachimGoedhart/Shiny_step-by-step

Running your app

- Via web: Shinyapps.io
- Via web: Shinyserver
- Local: needs R/Rstudio

Limitations

- Upload limit (5 Mb?)
- Connection loss
- Preferences/data not stored