Plotting Part 2.0

Shiny

The R Bootcamp
Twitter: @therbootcamp

September 2017

Shiny

*What is the Matrix (**Shiny**)?*

*No one can be told what the Matrix (**Shiny**) is… You have to see it for yourself.*

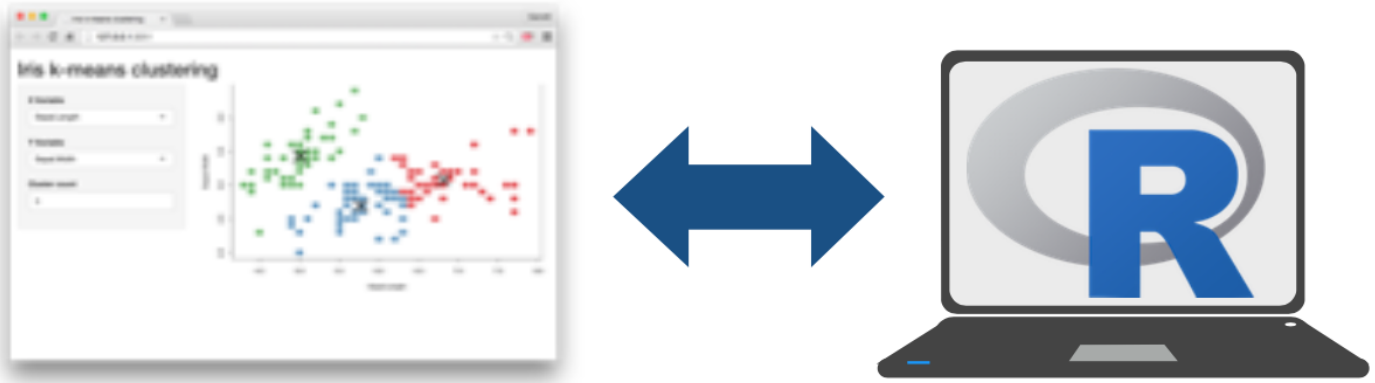
This is Shiny

`Shiny (https://shiny.rstudio.com/)`

from (https://www.rstudio.com/) Back to Gallery (https://shiny.rstudio.com/gallery/) Get Code (https://github.com/rstudio/shiny-e

What is Shiny?

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)



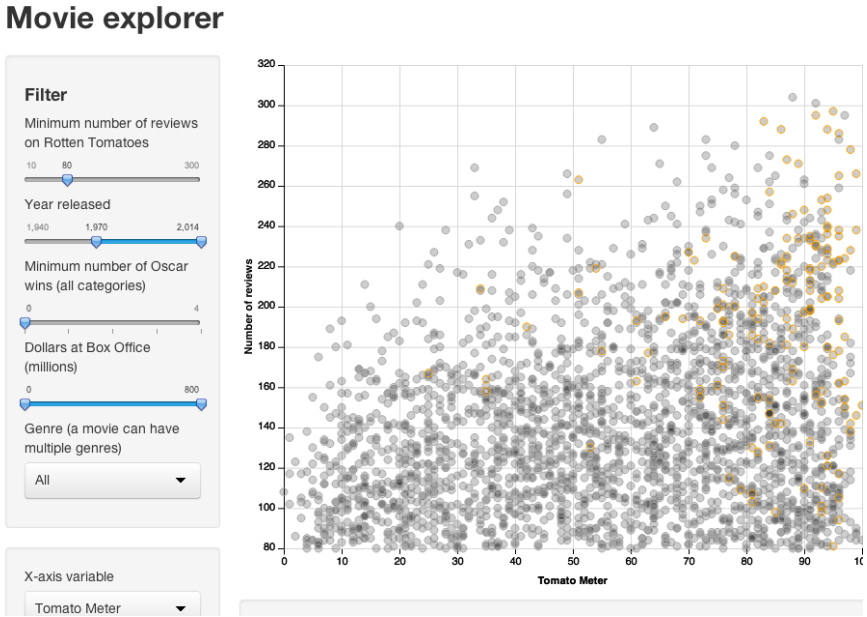
Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

R Studio

What does that mean?

- Shiny allows you to easy create interactive websites to explore, analyse, and visualize data
- No need to learn HTML, CSS, Javascript

I can easily do this all in Shiny!



Histogram Example

Balloon Analogue Risk Task

Please Wait

FFTrees Example

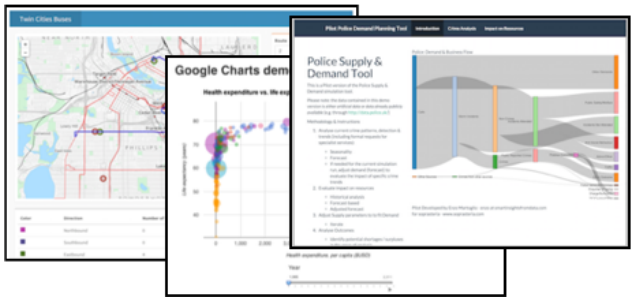
Please Wait

GlioVis

Shiny

What we will cover today

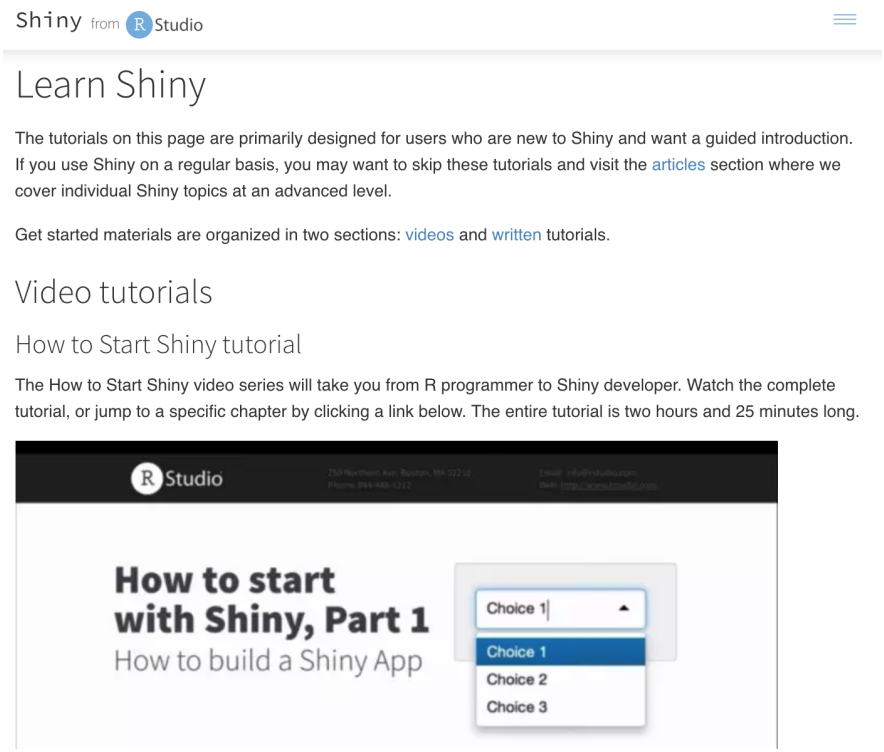
What you can DO with shiny



R Studio

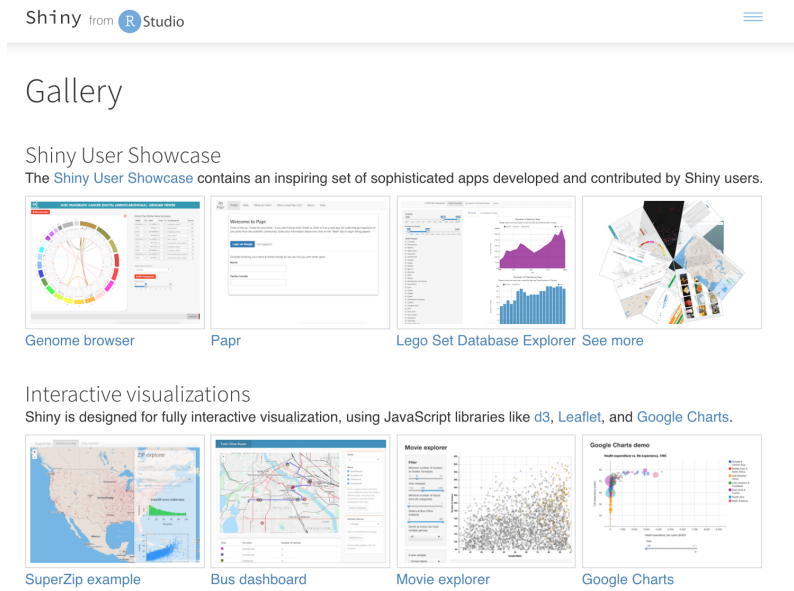
Shiny Tutorials

R Studio has great tutorials for creating Shiny Apps



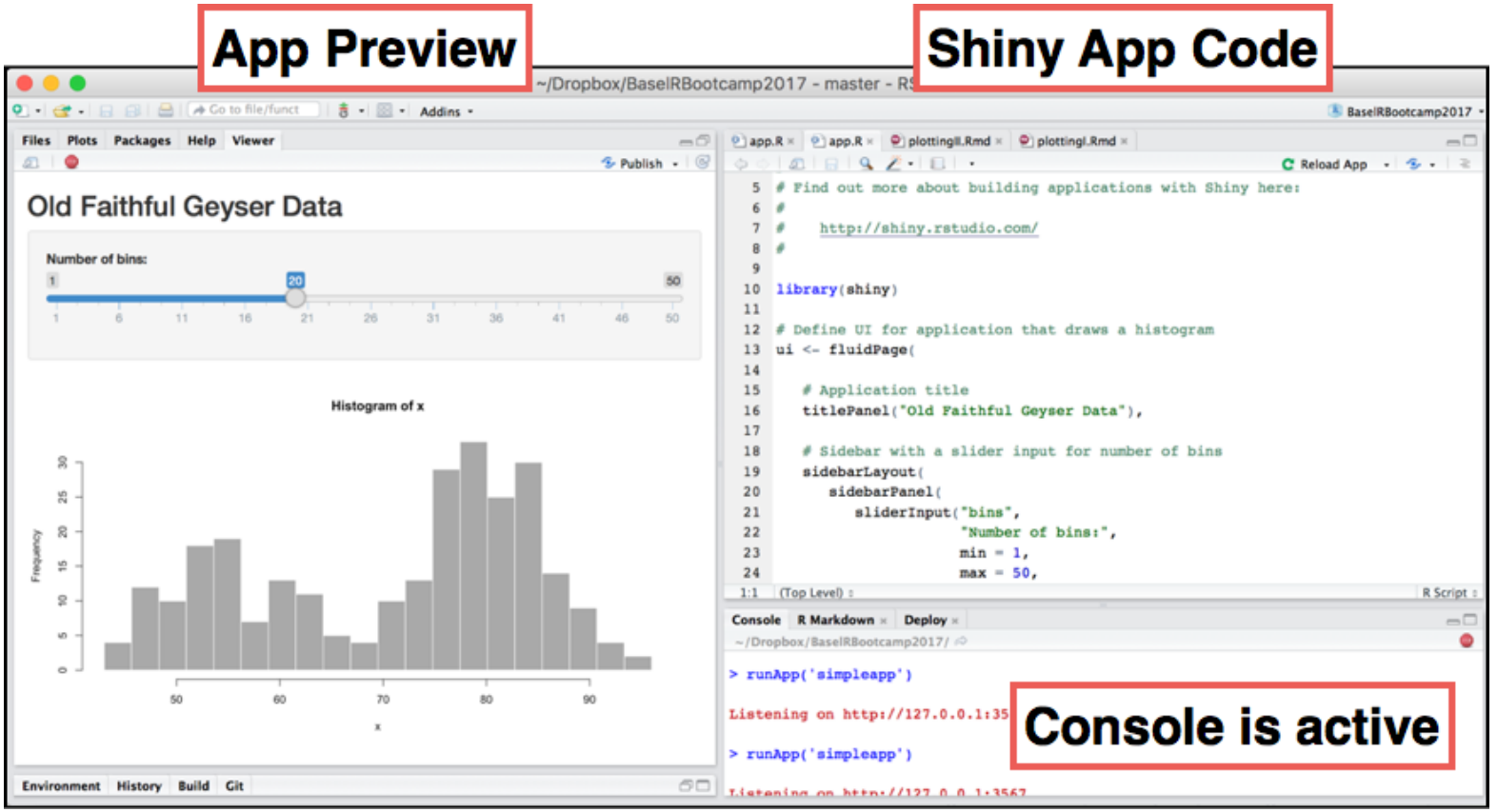
R Studio

Learn by example. Tons of shiny apps online. Most code is available.



R Studio

How programming a Shiny App looks



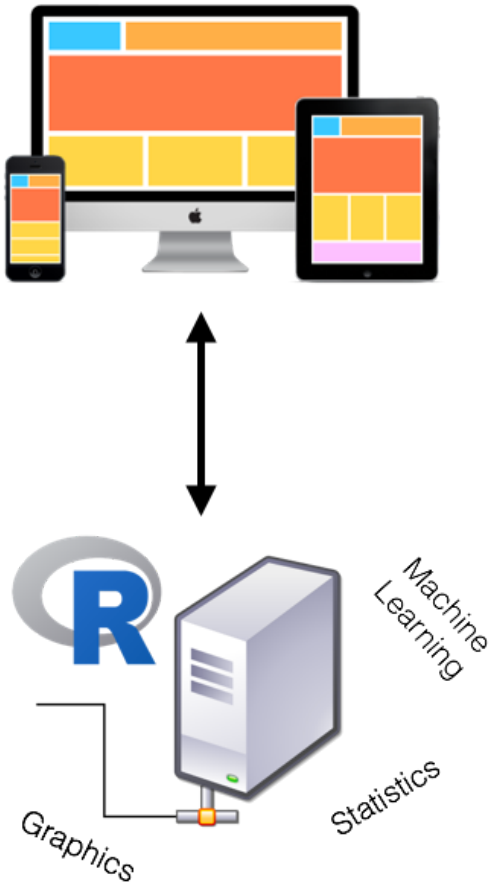
Structure of a Shiny App

User Interface: **ui()**

```
ui <- (
# Set overall layout
# Add widgets,
# Display plots
)
```

R Server: **server()**

```
server <- function(input, output) {
# Access Data
# Run analyses
# Create and render plots
)
}
```



Let's explore the user interface of an app!

P.S. You'll create this app in the practical!

User Interface

User Interface

User Interface

User Interface

User Interface

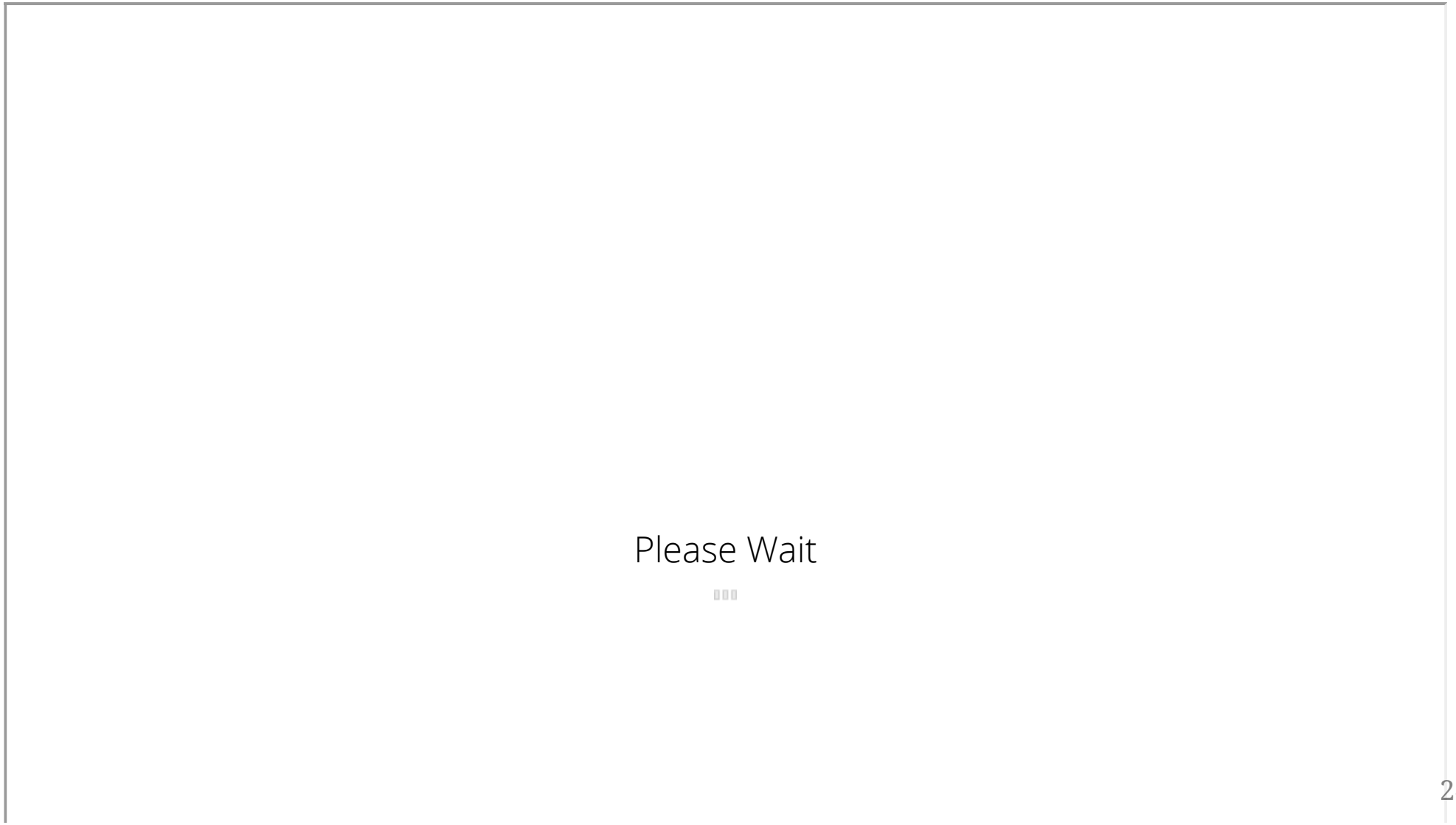
User Interface

User Interface

User Interface

User Interface

The final app!



User Interface

The user interface typically contains two main components: Widgets and

User Interface, Widgets

- Widgets are simple fields added to the user interface for users to add inputs.

User Interface, Layout

- You can control the layouts of apps with layout functions

Server

- All R code that creates plots, does machine learning, accesses databases, searches twitter data (really anything!) goes in the `server()` function.

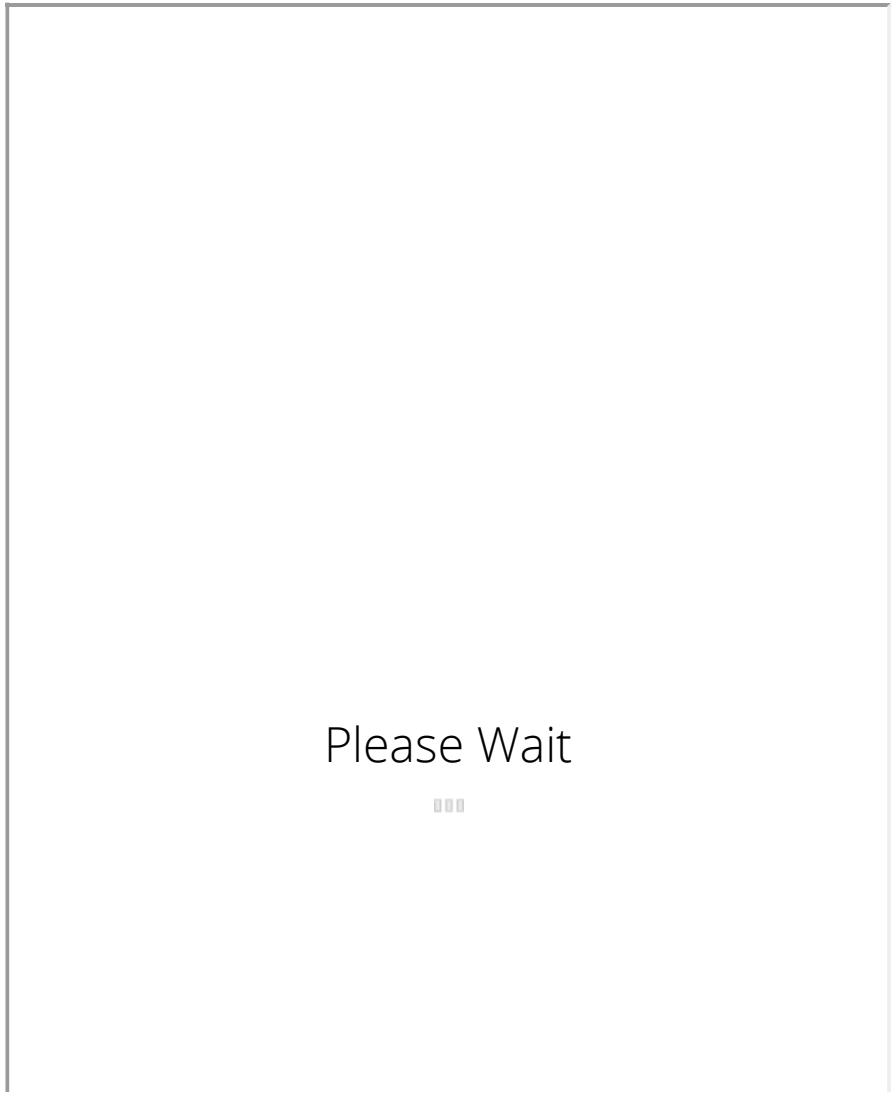
Server

> How does the server communicate with the user interface?

To send output to the user interface, you must use a render function

- To present output (e.g.; a plot), it must be *rendered* in the server using a special rendering function `renderXX()`,
- Once it is rendered, it is sent to the `output` and displayed in the user interface using an `xxOutput()` function.

Server




```
library(shiny)

# User Interface:
ui <- fluidPage(
  mainPanel(
    textInput("Title", "Title"),
    plotOutput("myplot")  # Create the plot
  )
)

server <- function(input, output) {

  # Define x
  x <- ChickWeight$weight

  # Send rendered plot to output
  output$myplot <- renderPlot({
    hist(x, main = input$Title)
  })
}

shinyApp(ui = ui, server = server)
```

Rendering output

The Shiny cheatsheet explains the most common functions for rendering and presenting output.

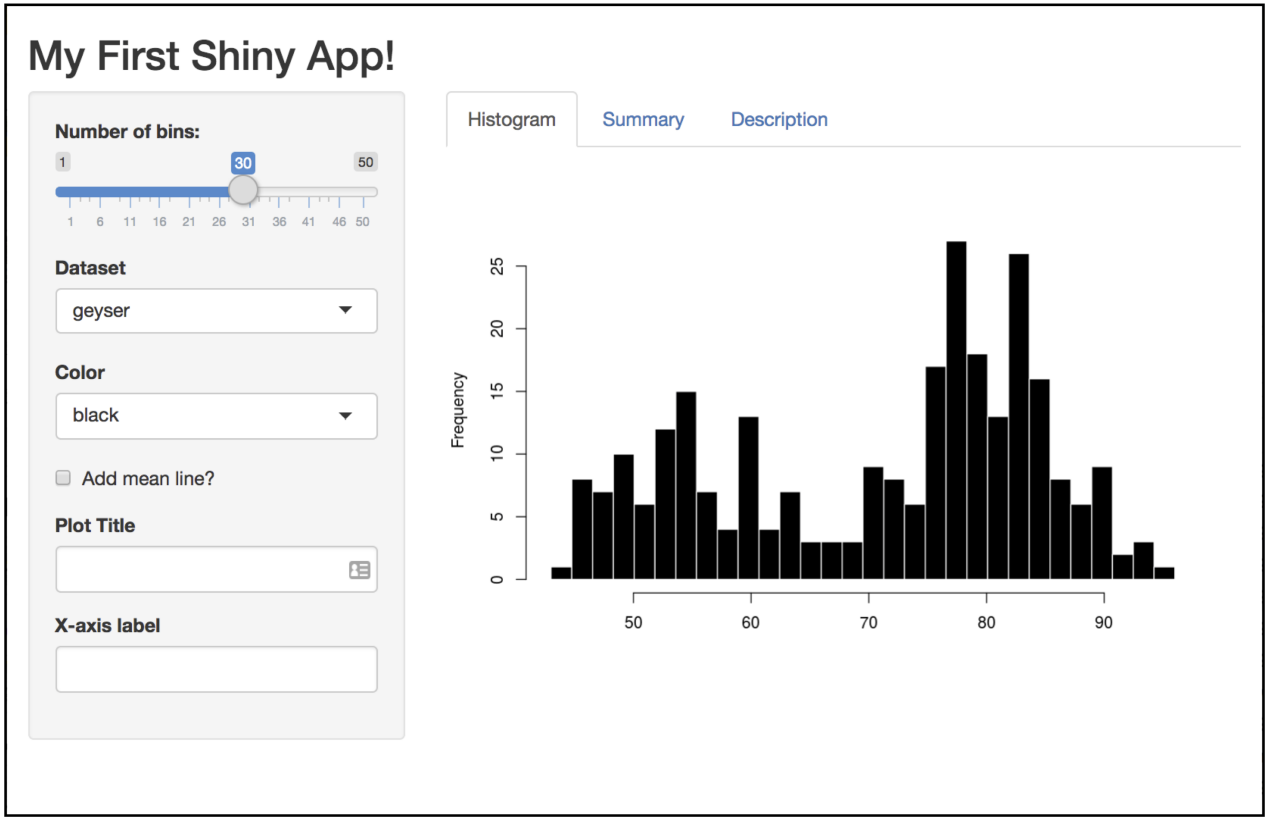
Publishing (hosting) an app

R Studio

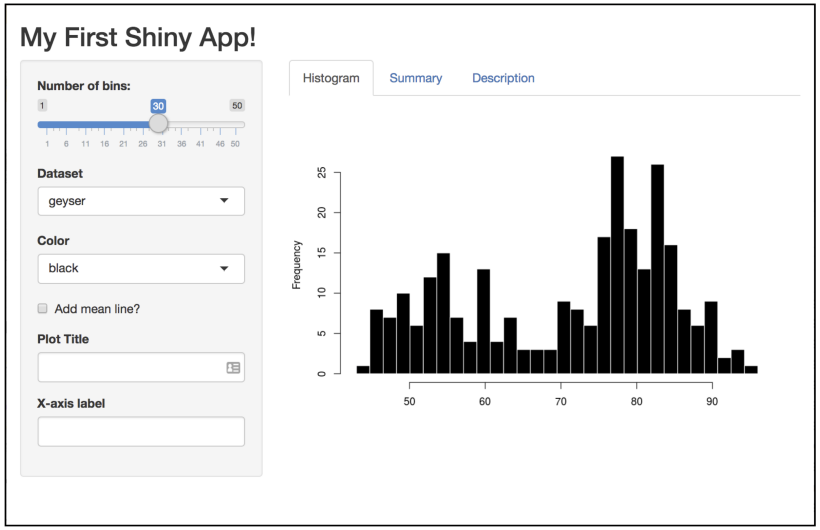
- You can always run a Shiny app locally on your machine.
- To get it online, you need to put it on a Shiny Server.
- Publish an app (with some restrictions) at http://shinyapps.io from RStudio with one click!
- Install a local server at your business (RStudio: $10,000 / year)
- Other providers exist (e.g.; Amazon Web Services)

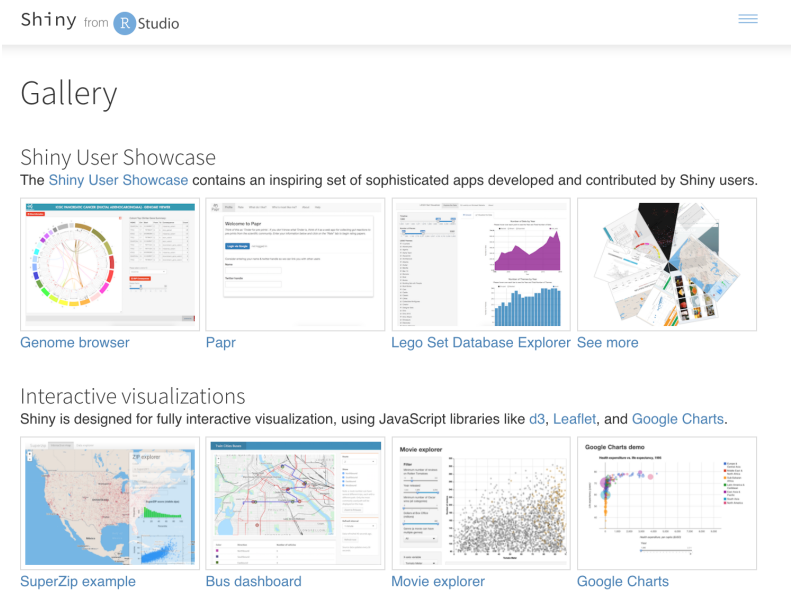
Practical

- In the practical, we will create the following app from scratch, and publish it online!



Questions?





R Studio

Plotting II Pratical

Link to Plotting II practical