

NC STATE UNIVERSITY

Basics of RShiny

Justin Post & Matt Beckman

What is R Shiny?

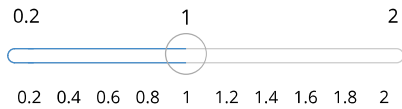
- R Shiny Package (<http://shiny.rstudio.com/>)
 - Developed by RStudio
 - Allows for creation of apps and dashboards
- Usually a `.R` file (or two) with special code to create an app
 - `ui.R` (User Interface)
 - `server.R` (R functions that run/respond to UI)
 - `app.R` (both UI and server combined)
- We'll just add them to HTML documents
- Requires no HTML, CSS, or JavaScript!

Example App

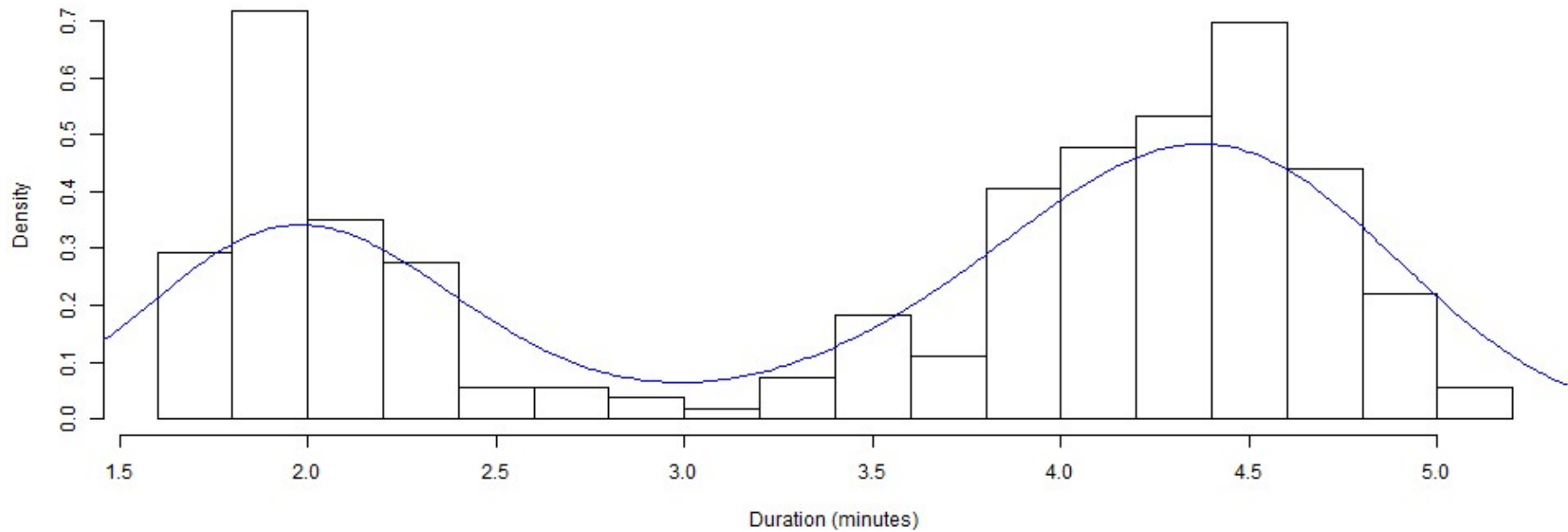
Number of bins:

20

Bandwidth adjustment:



Geyser eruption duration



Available Apps

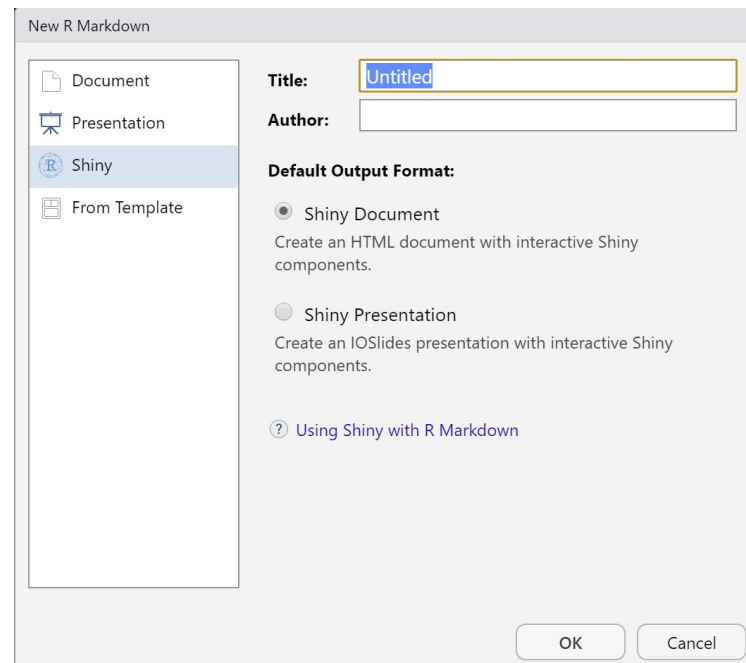
- Many available resources. Many have their source code available on github!
- Book of Apps for Statistics Teaching (BOAST) (<https://sites.psu.edu/shinyapps/>)
- Stat Concepts (<https://github.com/gastonstat/shiny-introstats/>)
- More Stat Concepts (https://www.researchgate.net/publication/298786680_Web_Application_Teaching_Tools_for_Statistics_Using_R_and_Shiny)
- Even more! (<http://www.statistics.calpoly.edu/shiny>)
- Shiny Gallery/Showcase (<https://shiny.rstudio.com/gallery/>)

Our plan

- Learn about user interface (UI) elements
 - Input widgets (sliders, numeric inputs, etc.)
 - Formatting of text
 - UI layout
- Understand how the server (R) backend works with the UI elements
 - Accessing UI inputs
 - Creating outputs

Create a shiny markdown doc

- File -> New file -> R Markdown
- Really, we just need to add `runtime: shiny` to the YAML header for an HTML doc!



Adding Widgets

- Widgets can be added using their `*Input` functions
- You can just place widgets within an R code chunk!

Button

Action

`actionButton()`

Single checkbox

☒ Choice A

`checkboxInput()`

Checkbox group

☒ Choice 1
☐ Choice 2
☐ Choice 3

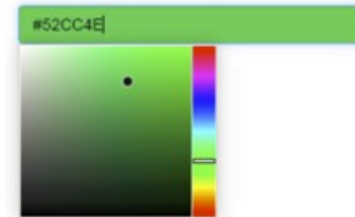
`checkboxGroupInput()`

Date input

2014-01-01

`dateInput()`

Colour input



`colourpicker::colourInput()`

Date range

2014-01-24 to 2014-01-24

`dateRangeInput()`

File input

Choose File No file chosen

`fileInput()`

Numeric input

1

`numericInput()`

Password Input

.....

`passwordInput()`

Radio buttons

☒ Choice 1
☐ Choice 2
☐ Choice 3

`radioButtons()`

Select box

Choice 1

`selectInput()`

Sliders



`sliderInput()`

Text input

Enter text...

`textInput()`

Text area

Multiple lines
of text

`textAreaInput()`

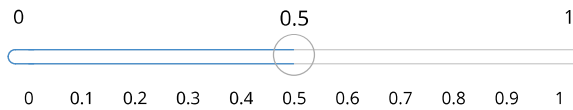
Widget Example

```
numericInput("num", "Enter a Number", value = 0, min = 0, max = 100)
```

```
sliderInput("slide", label = "A Slider!", min = 0, max = 1, value = 0.5, step = 0.05)
```

Enter a Number

A Slider!



Adding Formatted Text

Can also add:

- Any plain strings
- Formatted text (using HTML type functions)

shiny function HTML5 equivalent creates

p	<p>	A paragraph of text
h1	<h1>	A first level header
h2	<h2>	A second level header
h3	<h3>	A third level header
h4	<h4>	A fourth level header
h5	<h5>	A fifth level header
h6	<h6>	A sixth level header
a	<a>	A hyper link
br	 	A line break (e.g. a blank line)

div	<div>
span	
pre	<pre>
code	<code>
img	
strong	
em	
HTML	

A division of text with a uniform style

An in-line division of text with a uniform style

Text 'as is' in a fixed width font

A formatted block of code

An image

Bold text

Italicized text

Directly passes a character string as HTML code

Widget & Text Example

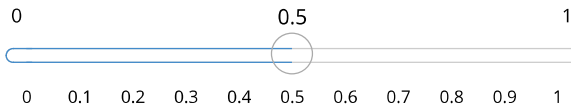
```
h2("First App title!")  
a("RStudio link", href = "https://www.RStudio.com")  
numericInput("num", "Enter a Number", value = 0, min = 0, max = 100)  
sliderInput("slide", label = "A Slider!", min = 0, max = 1, value = 0.5, step = 0.05)
```

First App title!

RStudio link (<https://www.RStudio.com>)

Enter a Number

A Slider!



Formatting

- `inputPanel()` allows you to add user inputs, text, etc. in a single row
- Syntax:

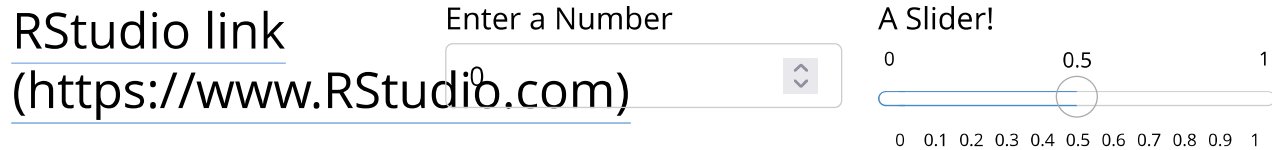
```
inputPanel(  
  widgetName1(...),  
  textFormatting(...),  
  widgetName2(...),  
)
```

Example

```
h2("First App title!")
```

```
inputPanel(  
  a("RStudio link", href = "https://www.RStudio.com"),  
  numericInput("num", "Enter a Number", value = 0, min = 0, max = 100),  
  sliderInput("slide", label = "A Slider!", min = 0, max = 1, value = 0.5, step = 0.05)  
)
```

First App title!



The screenshot displays the user interface of an R Shiny application. It features three distinct input components arranged horizontally. On the left, a text label 'RStudio link' is positioned above a text input field containing the URL 'https://www.RStudio.com'. The middle component is a numeric input field with the label 'Enter a Number' and a value of '0'. On the right, a slider input is labeled 'A Slider!' and shows a range from 0 to 1 with a current value of 0.5. The slider has major tick marks at 0, 0.5, and 1, and minor tick marks every 0.1 units.

Creating Outputs

- Outputs can be created using their `render*` functions

Rendering functions

Functions that you use in your application's server side code, assigning them to outputs that appear in your user interface.

<code>renderPlot</code>	Plot Output
<code>renderText</code>	Text Output
<code>renderPrint</code>	Printable Output
<code>renderDataTable</code>	Table output with the JavaScript library DataTables
<code>renderImage</code>	Image file output
<code>renderTable</code>	Table Output
<code>renderUI</code>	UI Output
<code>downloadHandler</code>	File Downloads
<code>reactivePlot</code>	Plot output (deprecated)
<code>reactivePrint</code>	Print output (deprecated)
<code>reactiveTable</code>	Table output (deprecated)
<code>reactiveText</code>	Text output (deprecated)
<code>reactiveUI</code>	UI output (deprecated)

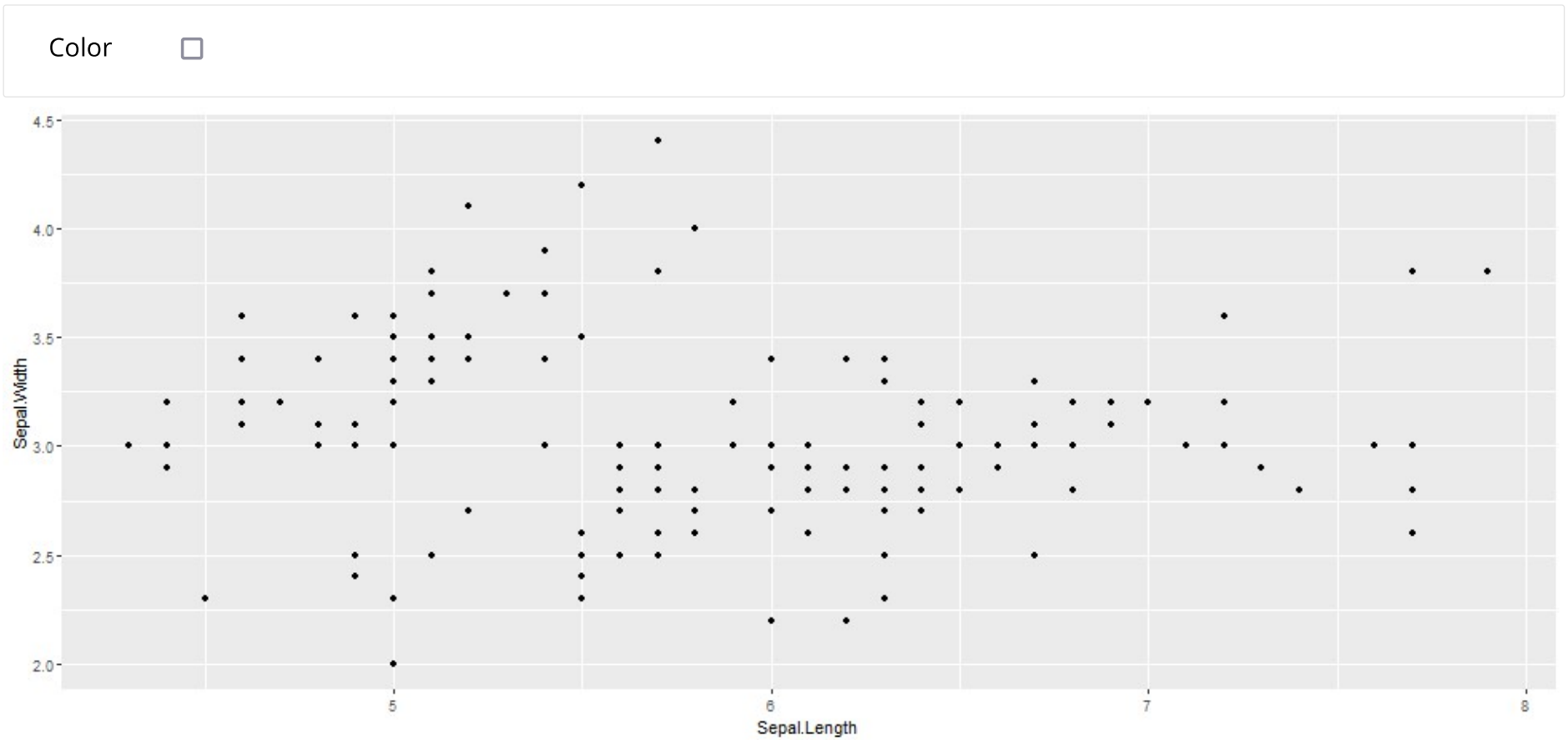
More About Widgets

- Widgets all follow the same structure
- `widgetName("internalID", label = "Title the user sees", ...)`
- The `internalID` is how you access the inputs when creating plots, summaries, etc.

Plot Example

```
inputPanel(  
  checkboxInput("addColor", "Color")  
)  
  
renderPlot({  
  g <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width))  
  if(input$addColor){  
    g + geom_point(aes(color = Species))  
  } else {  
    g + geom_point()  
  }  
})
```

Plot Example



Quick Sampling Distribution Example

Code to create some samples

- rows 1:n represents a sample of size n
- each column represents a data set

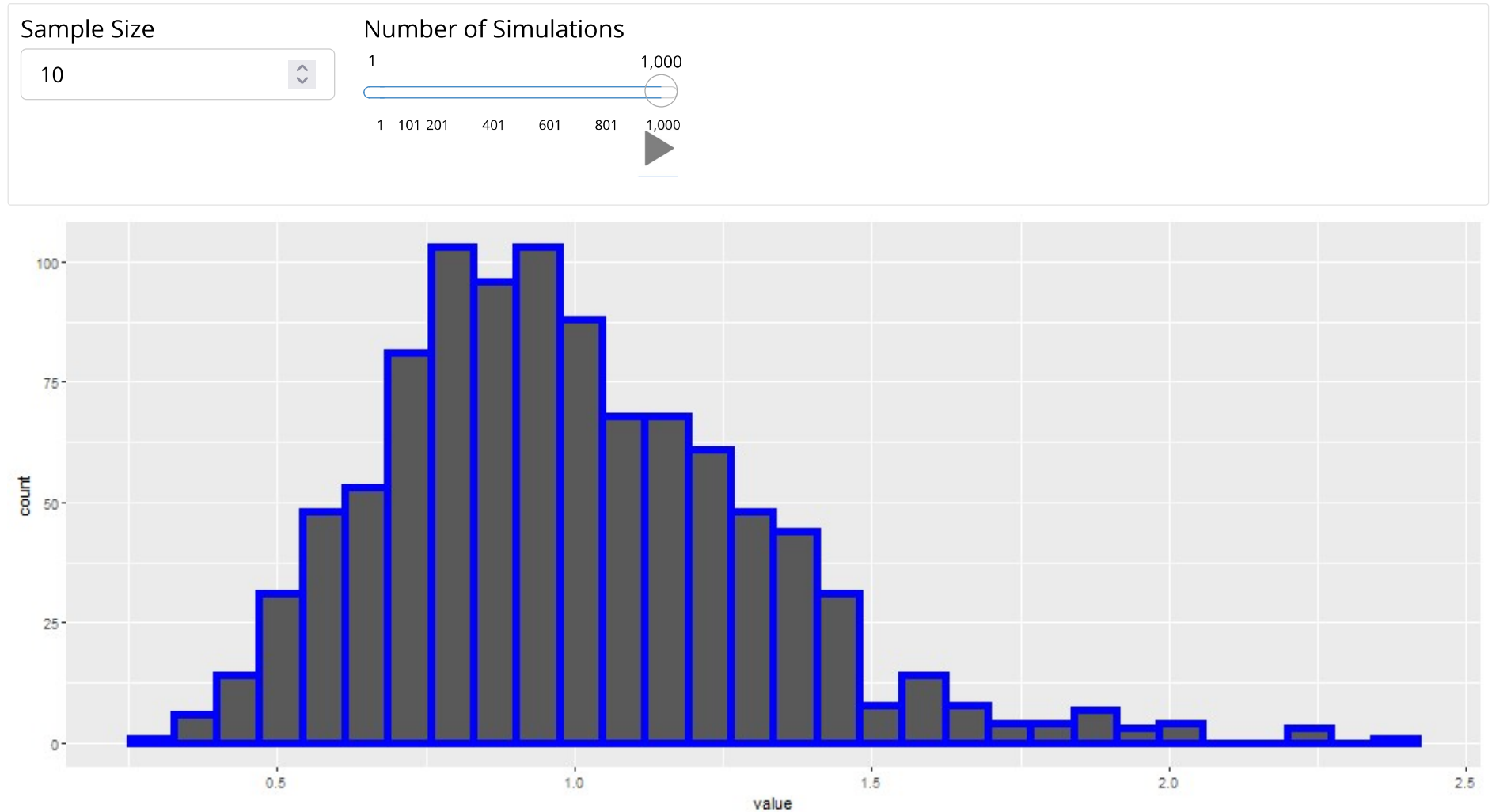
```
simData <- replicate(1000, rexp(n = 50, rate = 1)) %>%  
  as_tibble()  
simData
```

```
## # A tibble: 50 x 1,000  
##       V1      V2      V3      V4      V5      V6      V7      V8      V9      V10     V11     V12  
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2.08  1.39  0.542  1.49  0.817  3.63  3.54  0.157  0.506  0.454  0.427  3.50  
## 2 0.980 0.493 0.0540 1.96  0.194  0.674  1.30  0.195  0.448  1.56  0.120  0.159  
## 3 0.921 0.138 1.95   0.102  0.107  1.57  1.13  2.89  0.218  1.26  0.269  1.45  
## 4 1.41  0.511 4.73   0.286  0.0978 1.09  3.21  1.09  4.35  0.925  0.780  1.96  
## 5 0.592 0.204 0.722  0.773  0.214  0.763  0.871  0.174  0.0563 0.748  0.348  0.128  
## 6 0.981 2.25  0.130  0.0389 2.67   1.21  0.463  1.25  1.65  0.190  0.159  0.242  
## 7 0.208 1.49  0.899  0.0709 0.726  0.419  0.155  0.338  1.10  1.27  0.0378 0.0800  
## 8 0.868 0.567 0.0777 1.20   1.31  0.738  2.57  0.525  0.0991 0.186  0.824  2.02  
## 9 0.712 0.720 0.216  1.93   0.574  0.354  0.102  0.634  0.888  1.80  2.30  2.06  
## 10 1.33  0.653 2.12   0.451  2.61  0.962  0.479  0.378  2.38  0.680  1.22  1.50  
## # ... with 40 more rows, and 988 more variables: V13 <dbl>, V14 <dbl>,  
## #   V15 <dbl>, V16 <dbl>, V17 <dbl>, V18 <dbl>, V19 <dbl>, V20 <dbl>,
```

Distribution of \bar{Y} from a RS of Exp(1)

```
inputPanel(  
  numericInput("sampleSize", label = "Sample Size",  
    min = 2, max = 50, value = 10, step = 1),  
  sliderInput("numSamples", label = "Number of Simulations",  
    min = 1, max = 1000, value = 1, step = 1,  
    animate = animationOptions(interval = 325))  
)  
  
renderPlot({  
  ggplot(simData %>%  
    slice_head(n = input$sampleSize) %>%  
    select(1:input$numSamples) %>%  
    colMeans() %>%  
    as_tibble(),  
    aes(x = value)) +  
  geom_histogram(color = "Blue", size = 2)  
})
```

Distribution of \bar{Y} from a RS of Exp(1)



Much more to learn!

- Stand alone apps
- Shiny themes
- Dashboards
- UI Layouts
- Reactive contexts
- Dynamic UIs
- Hosting an app <https://www.shinyapps.io/> (shinyapps.io)