An Introduction to Data Visualization

Anamaria Crisan

@amcrisan acrisan@cs.ubc.ca http://cs.ubc.ca/~acrisan

A comment on *"How should we visualize data"*

There are two aspects of visualizations to think about:

How do you make a visualization?

tableau

Shiny
by RStudio

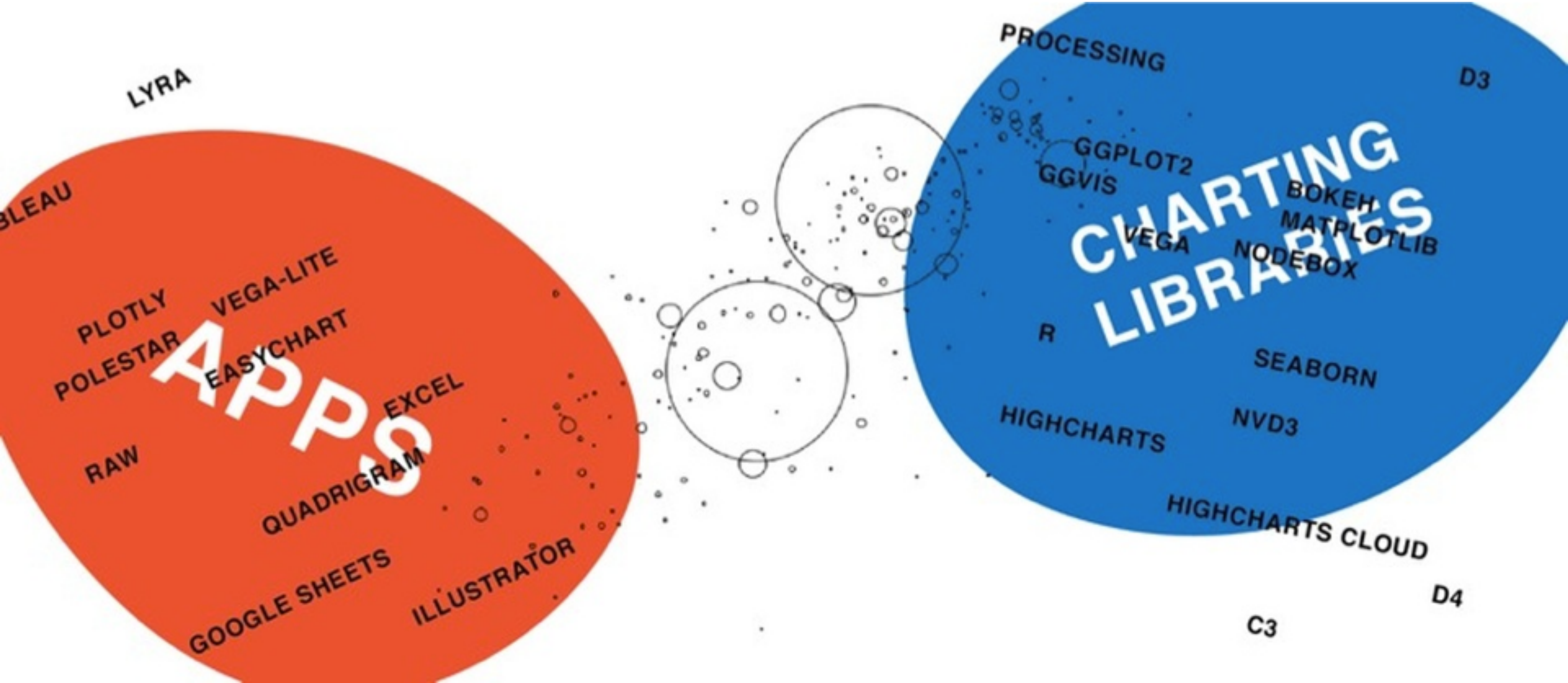
Is it the right visualization?

Data Visualization Tools

Tools & Libraries : An Overview

See this excellent post by Lisa Charlotte Rost : *http://bit.ly/2gRGx1J*

(figures taken from her post)

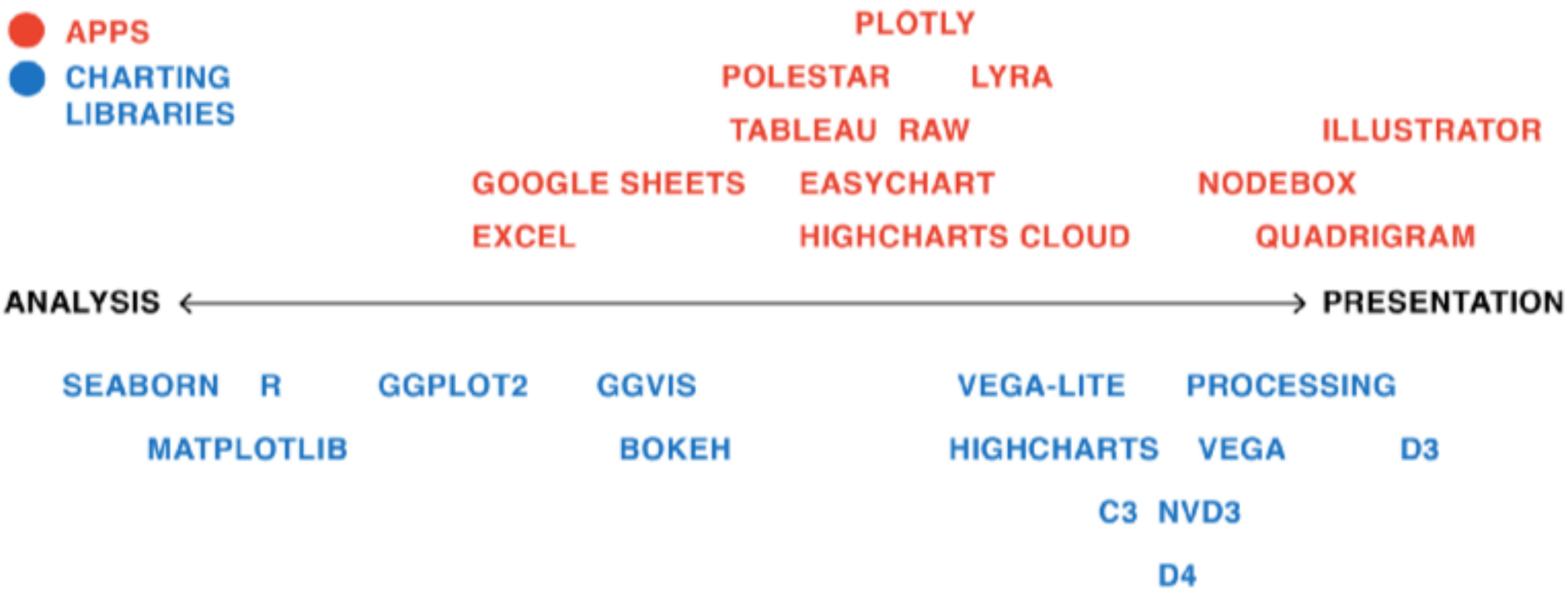


Tools & Libraries : An Overview

See this excellent post by Lisa Charlotte Rost : *http://bit.ly/2gRGx1J*

(figures taken from her post)

Analysis vs Presentation



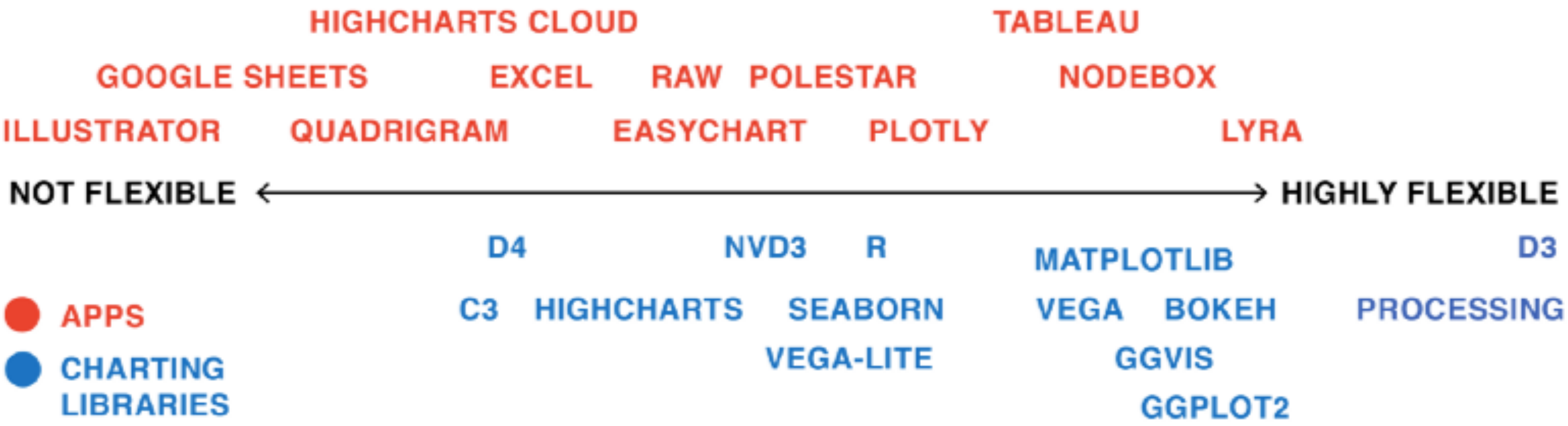
Tools & Libraries : An Overview

See this excellent post by Lisa Charlotte Rost : *http://bit.ly/2gRGx1J*

(figures taken from her post)

Extent of Flexibility

How easy/hard it is to make data visualizations (including custom/novel visualizations)

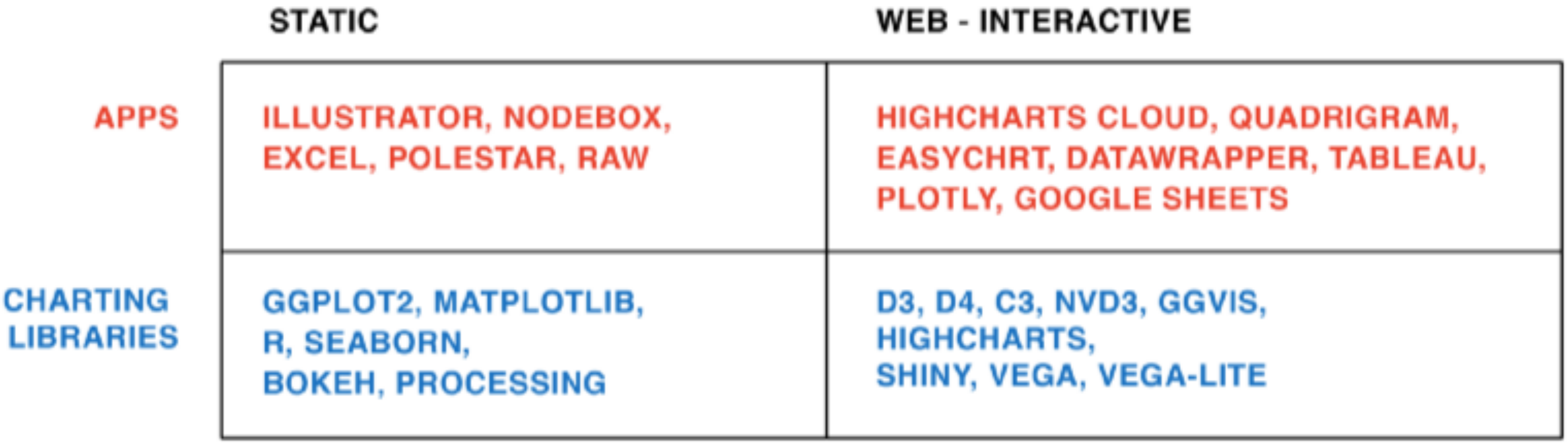


Tools & Libraries : An Overview

See this excellent post by Lisa Charlotte Rost : *http://bit.ly/2gRGx1J*

(figures taken from her post)

Static vs Interactive

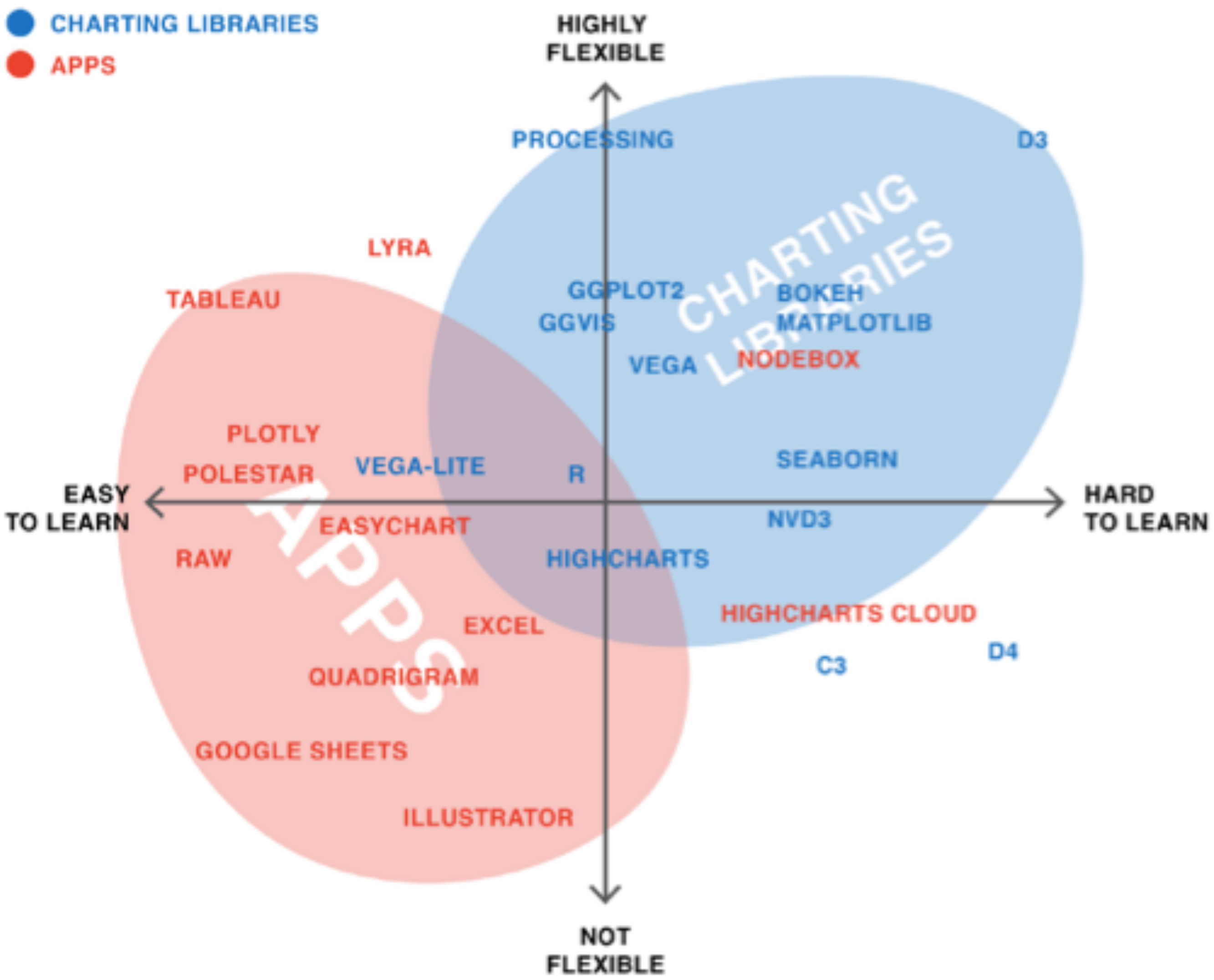


	STATIC	WEB - INTERACTIVE
APPS	ILLUSTRATOR, NODEBOX, EXCEL, POLESTAR, RAW	HIGHCHARTS CLOUD, QUADRIGRAM, EASYCHRT, DATAWRAPPER, TABLEAU, PLOTLY, GOOGLE SHEETS
CHARTING LIBRARIES	GGPLOT2, MATPLOTLIB, R, SEABORN, BOKEH, PROCESSING	D3, D4, C3, NVD3, GGVIS, HIGHCHARTS, SHINY, VEGA, VEGA-LITE

Tools & Libraries : An Overview

See this excellent post by Lisa Charlotte Rost : *http://bit.ly/2gRGx1J*

(figures taken from her post)



"There are no perfect tools, just good tools for people with certain goals"

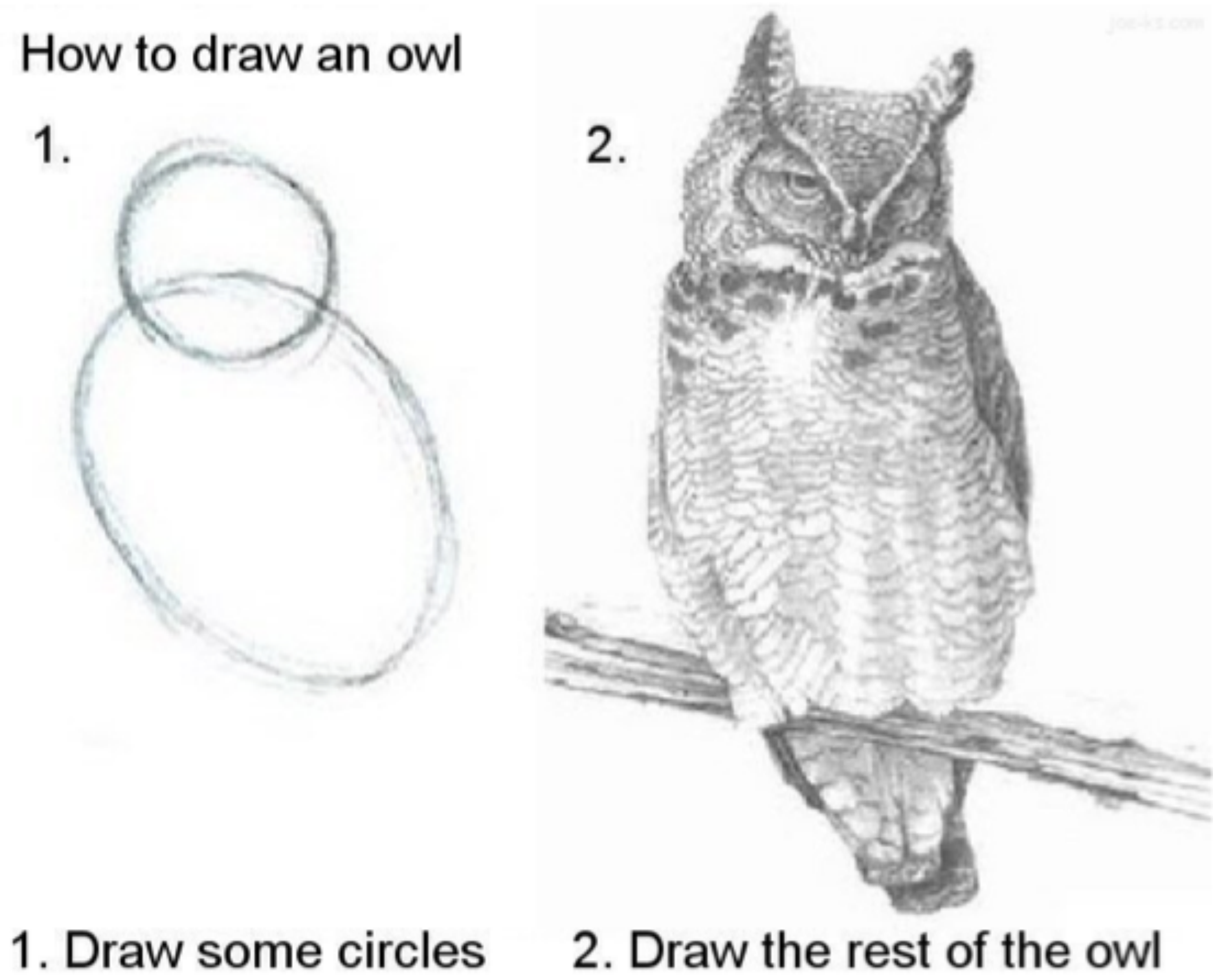
See a detailed table here: http://bit.ly/2DeWPwV

Shiny

Important Details on R and Shiny

- To introduce shiny, I am using selected slide content from several presentations developed for the "Shiny quick start guide" available here:
  - **Part 1:** bit.ly/shiny-quickstart-1
  - **Part 2:** bit.ly/shiny-quickstart-2
- These quick start slides are also presented as a video on the Shiny website
  - https://shiny.rstudio.com/tutorial/

The Challenges of Briefly Introducing Shiny



- Shiny is powerful and somewhat simpler than web languages - but it still takes some time to learn it
- I want to show you some basic info, and give you a sense of what Shiny can do - but there is a lot of in between I won't cover

What is Shiny?

Basic Structure

Presentation,
Layouts,
formatting

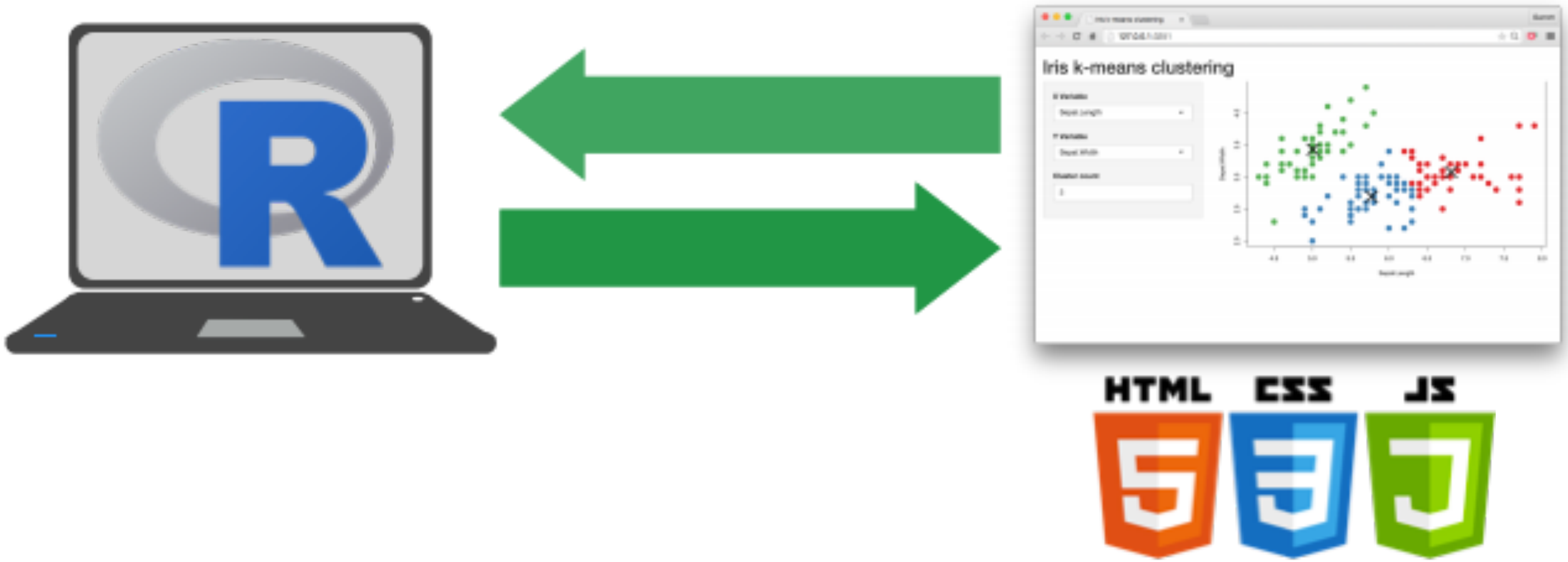
Webpage / Application
Behavior

+

Most glorious of
programming languages

How Does Shiny Work?

Every Shiny app is maintained by a computer running R

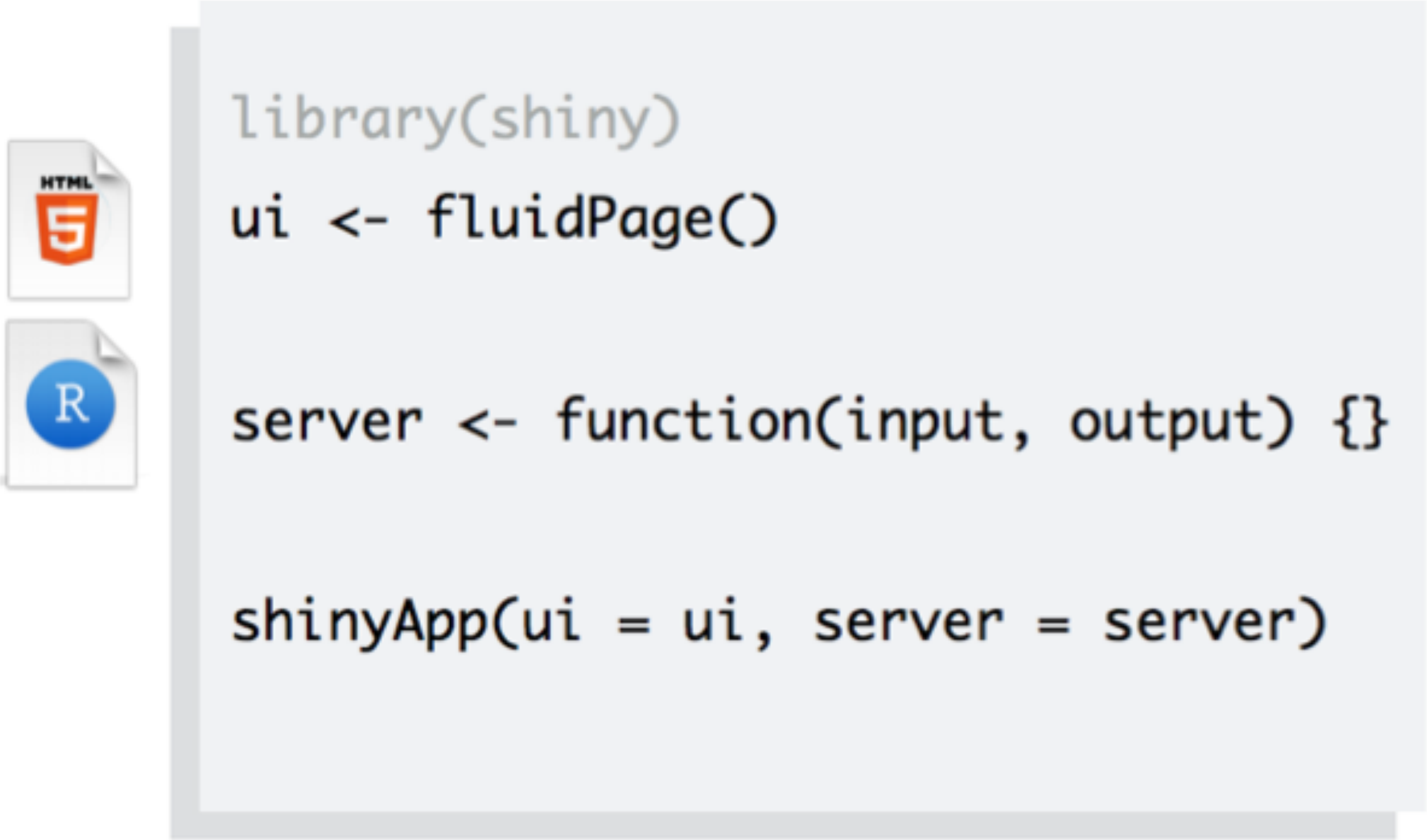


Server runs R code

User Interface (UI)

bit.ly/shiny-quickstart-1

Basic Structure of a Shiny Application

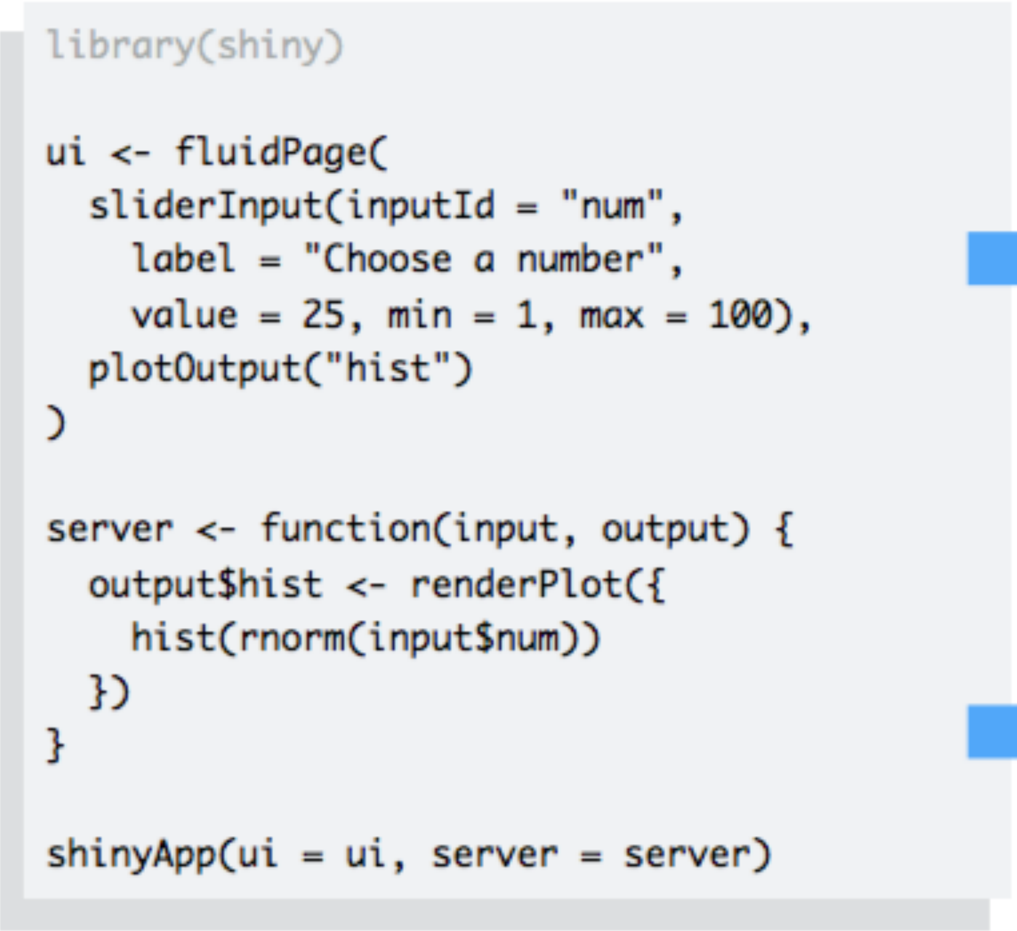


```
library(shiny)
ui <- fluidPage()


server <- function(input, output) {}


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

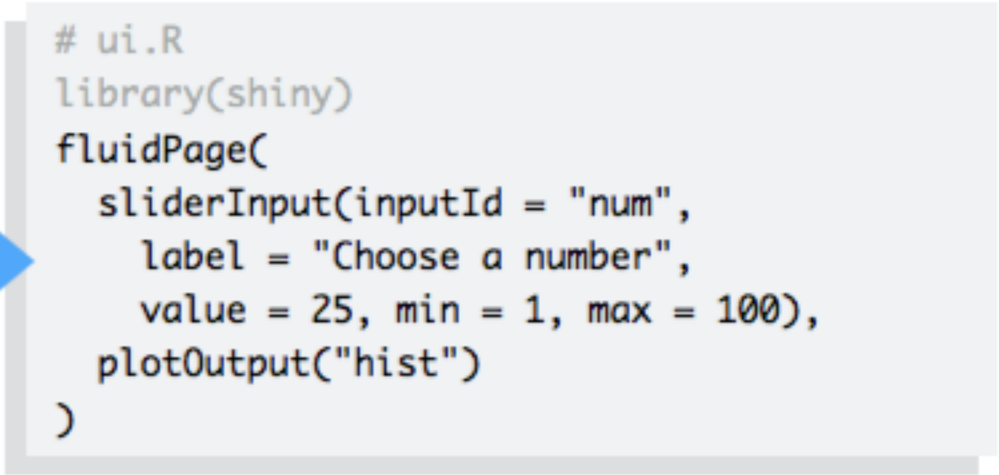
bit.ly/shiny-quickstart-1

Basic Structure of a Shiny Application

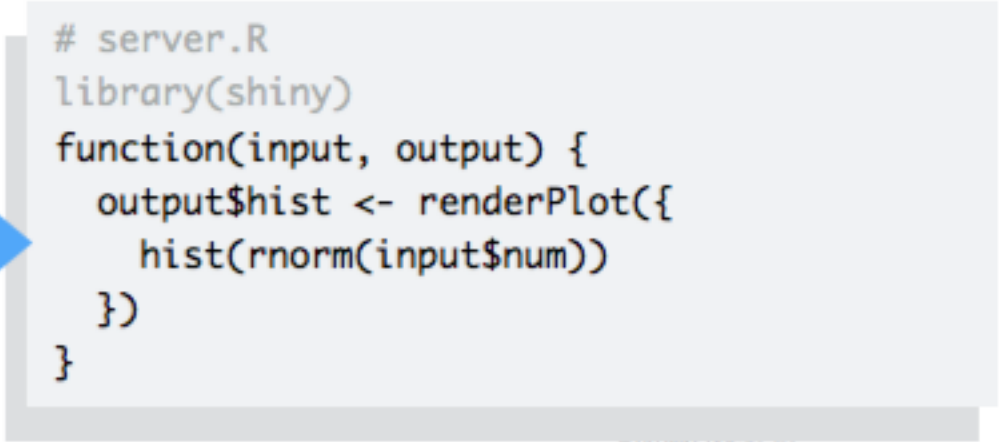
An entire application can be a single file, or broken down into a ui.R & server.R files



```r
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

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



```r
# ui.R
library(shiny)
fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)
```

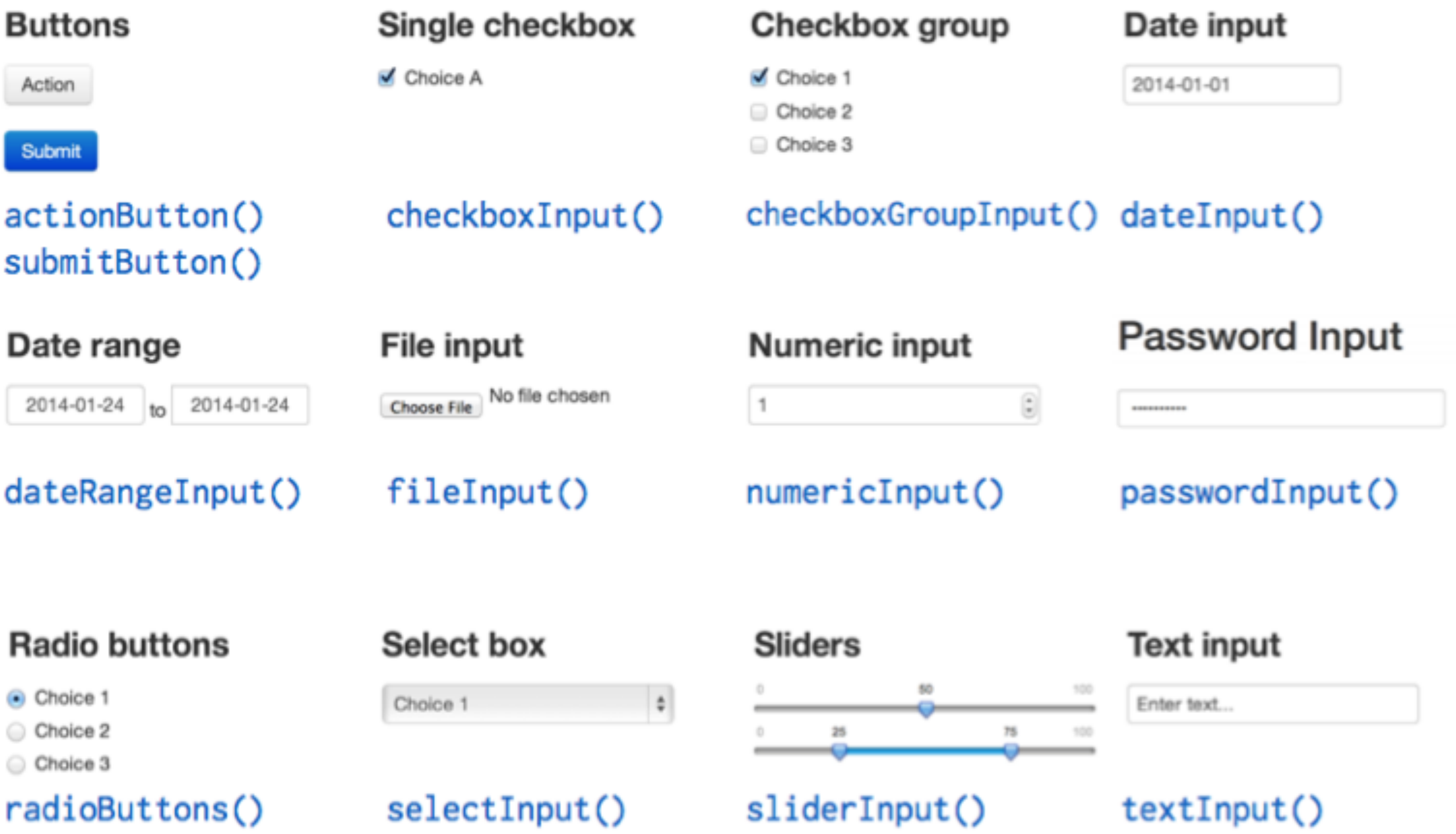


```r
# server.R
library(shiny)
function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}
```

You **must** use **app.r** (single file); or **ui.r** and **server.r** names, or your application won’t run

The user interface (ui.r) houses inputs & outputs

Example of inputs



Buttons

Action

Submit

`actionButton()`
`submitButton()`

Single checkbox

☑ Choice A

`checkboxInput()`

Checkbox group

☑ Choice 1
□ Choice 2
□ Choice 3

`checkboxGroupInput()`

Date input

2014-01-01

`dateInput()`

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()`

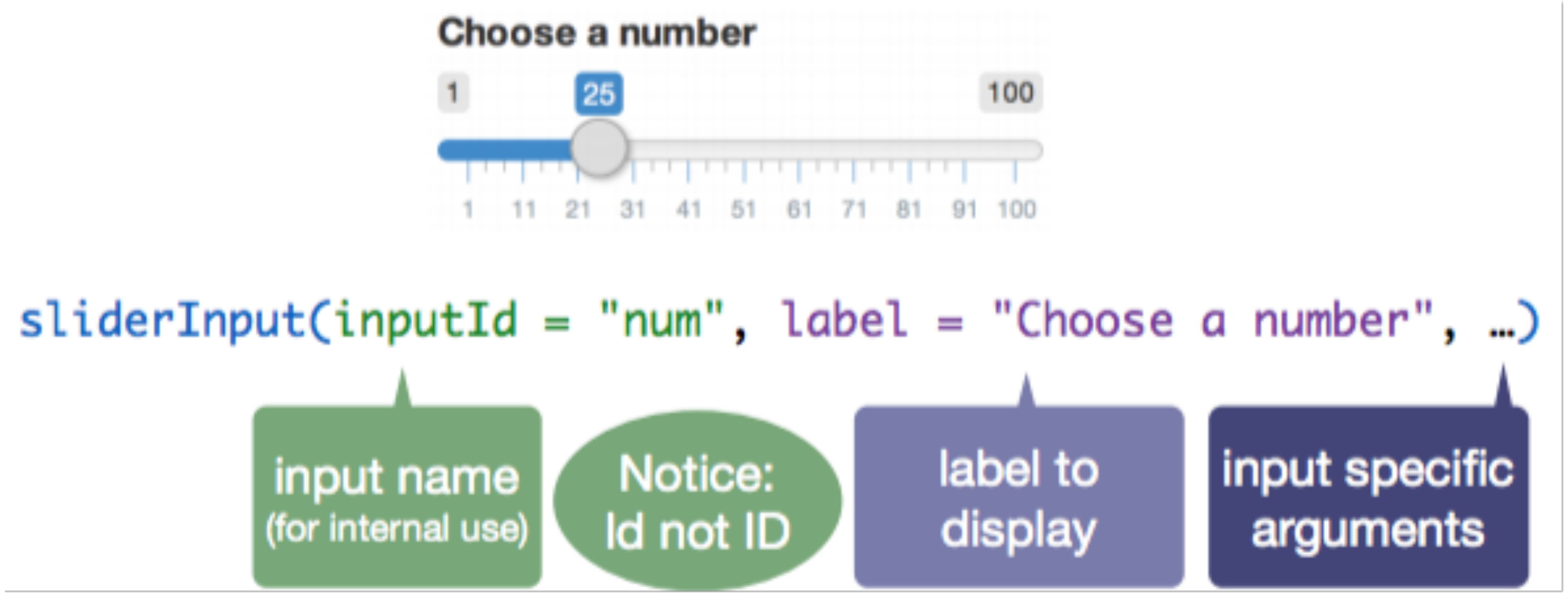
Text input

Enter text...

`textInput()`

The user interface (ui.r) houses inputs & outputs

Example of inputs



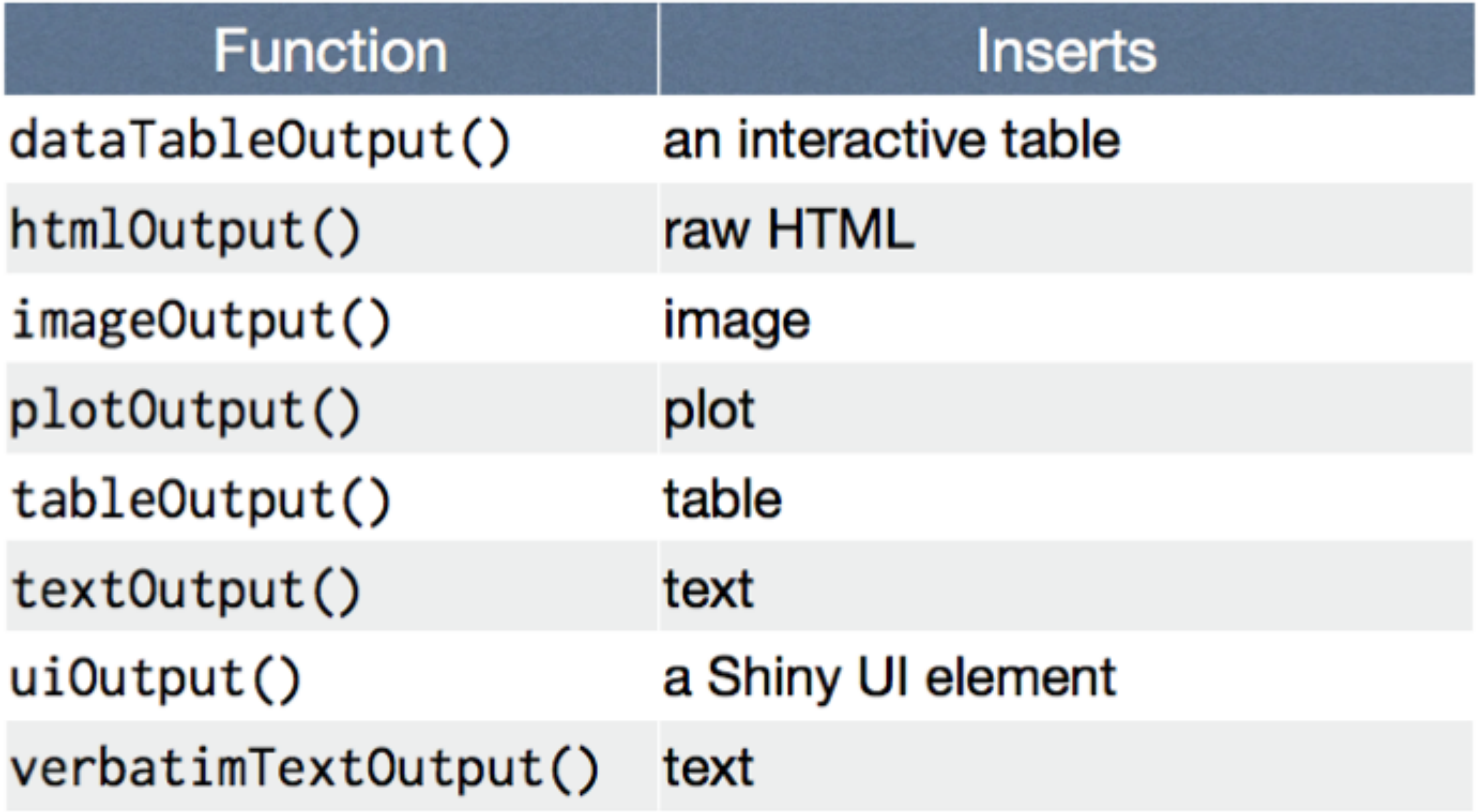
```
sliderInput(inputId = "num", label = "Choose a number", …)
```

Changing the slider number (i.e. dragging the circle left to right) causes R to automatically initiate an action to update all outputs that use the slider number.

bit.ly/shiny-quickstart-1

The user interface (ui.r) houses inputs & outputs

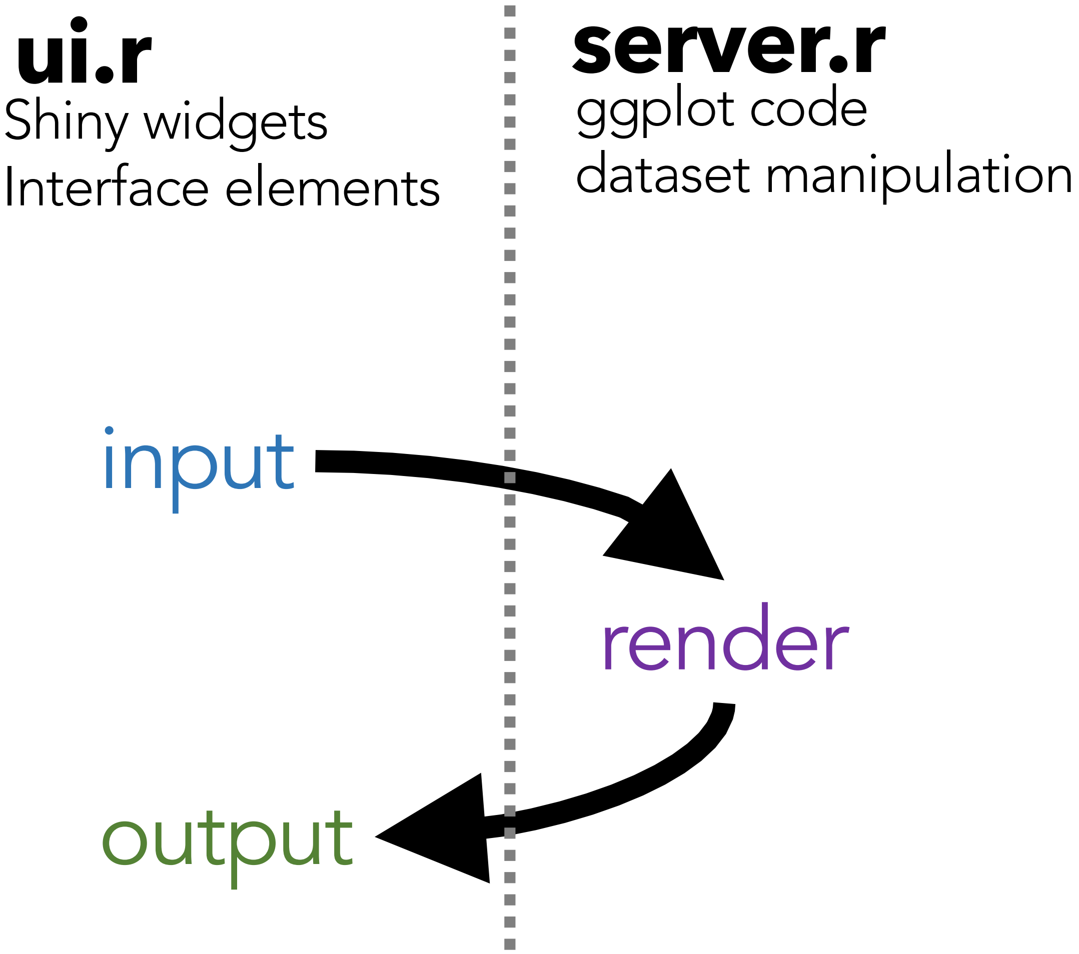
Example of outputs



Function	Inserts
dataTableOutput()	an interactive table
htmlOutput()	raw HTML
imageOutput()	image
● plotOutput()	plot
● tableOutput()	table
textOutput()	text
● uiOutput()	a Shiny UI element
verbatimTextOutput()	text

Function = command telling R & shiny to do some particular task

server.r stiches inputs & outputs together



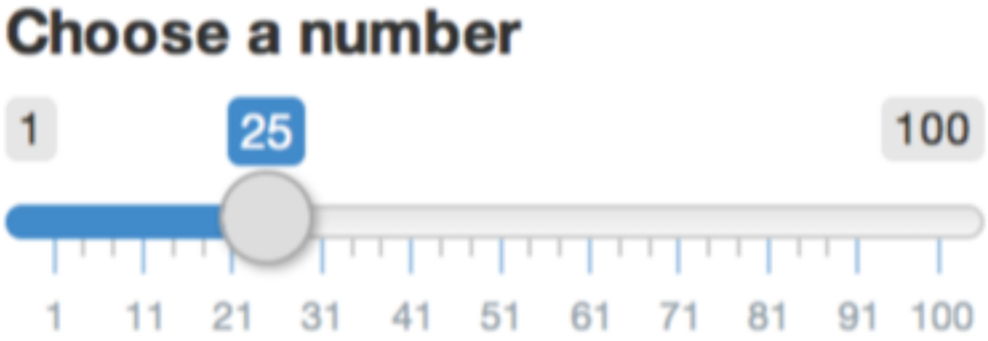
ui.r
Shiny widgets
Interface elements

server.r
ggplot code
dataset manipulation

Output & render functions are complimentary

output function	render function
plotOutput( )	renderPlot( )
tableOutput( )	renderTable( )
uiOutput( )	renderUI( )

A simple worked example

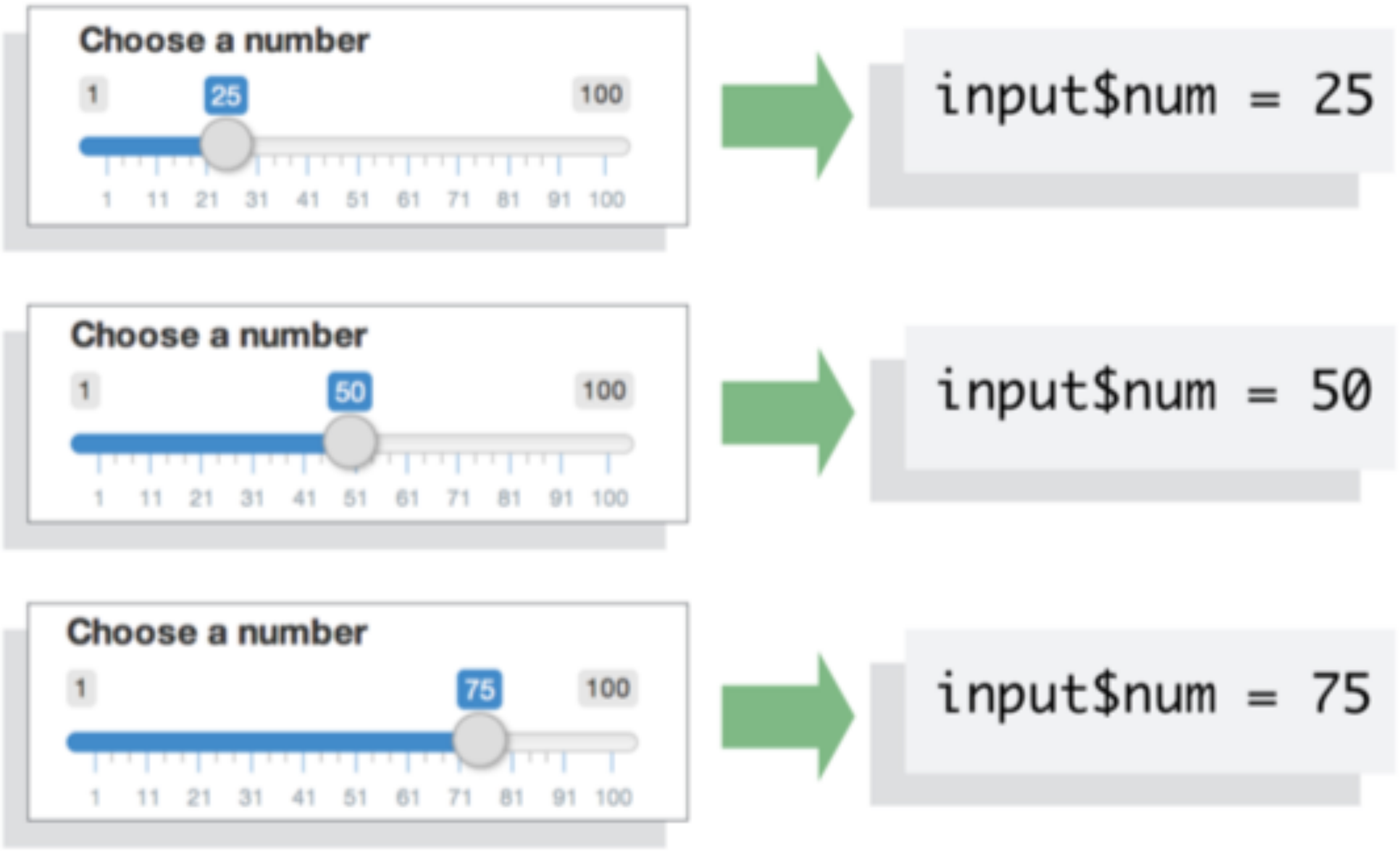


```
sliderInput(inputId = "num", label = "Choose a number", …)
```

this input will provide a value saved as **input$num**

A simple worked example

The input value changes whenever a user changes the input.



```
input$num = 25
```

```
input$num = 50
```

```
input$num = 75
```

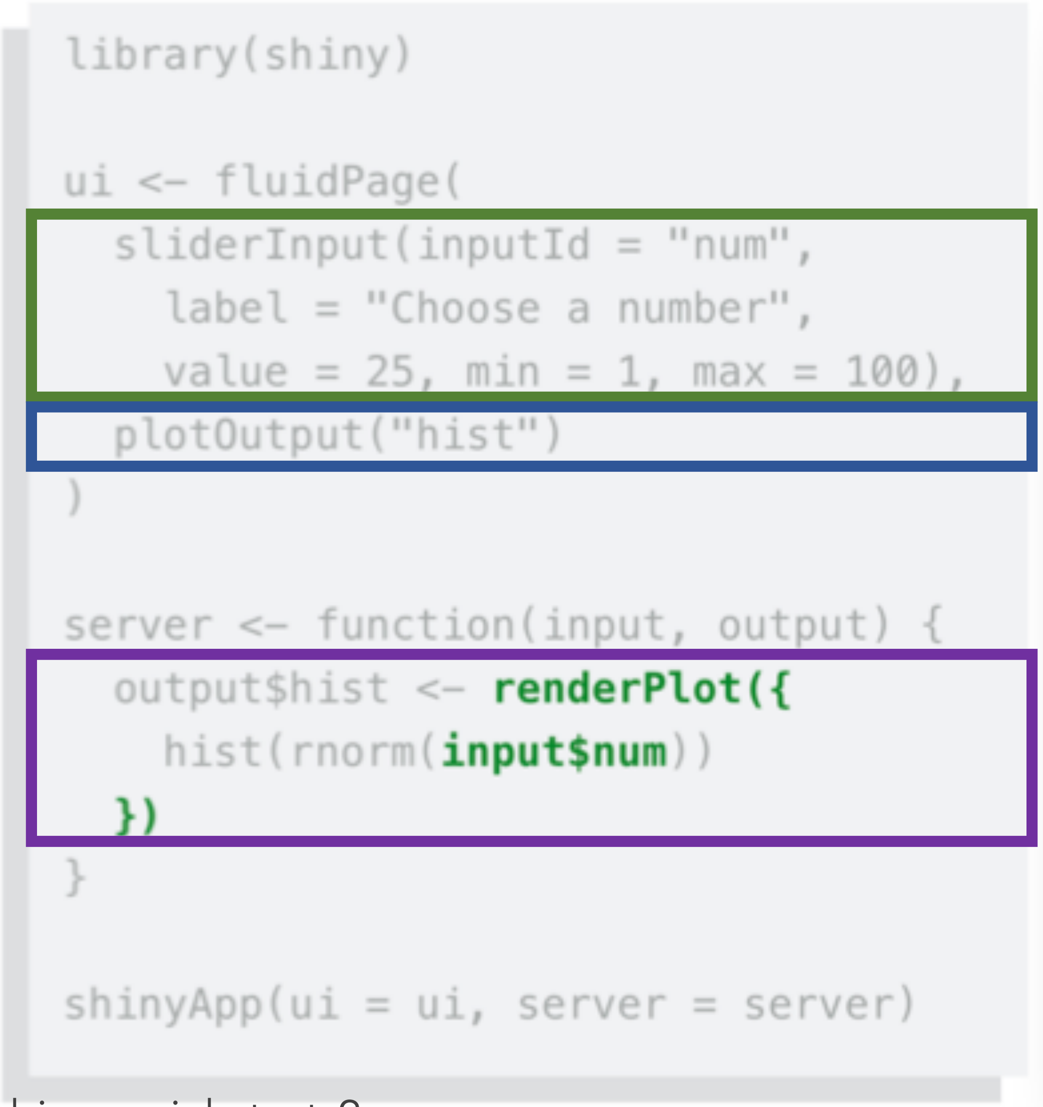
A simple worked example

Let’s say, when you change the number, you also want to change a histogram

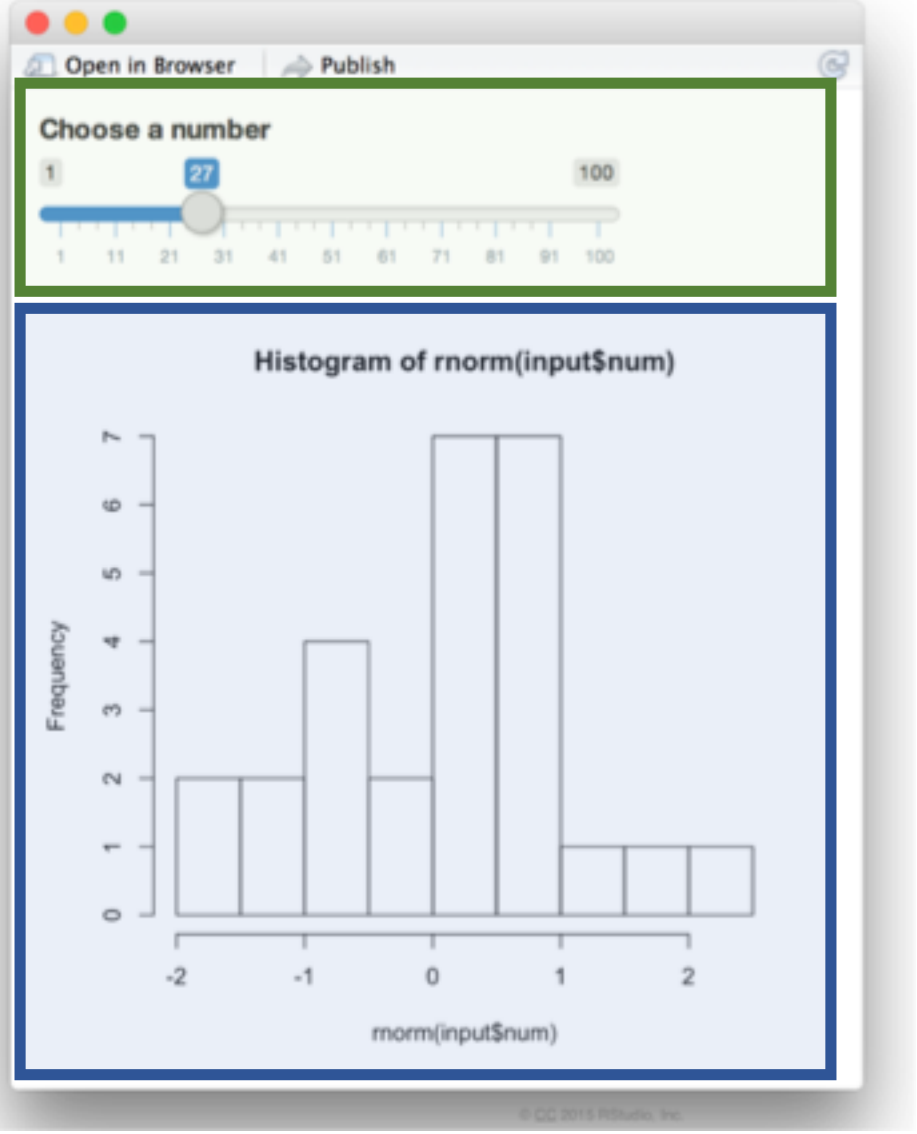
input

output

render



```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

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

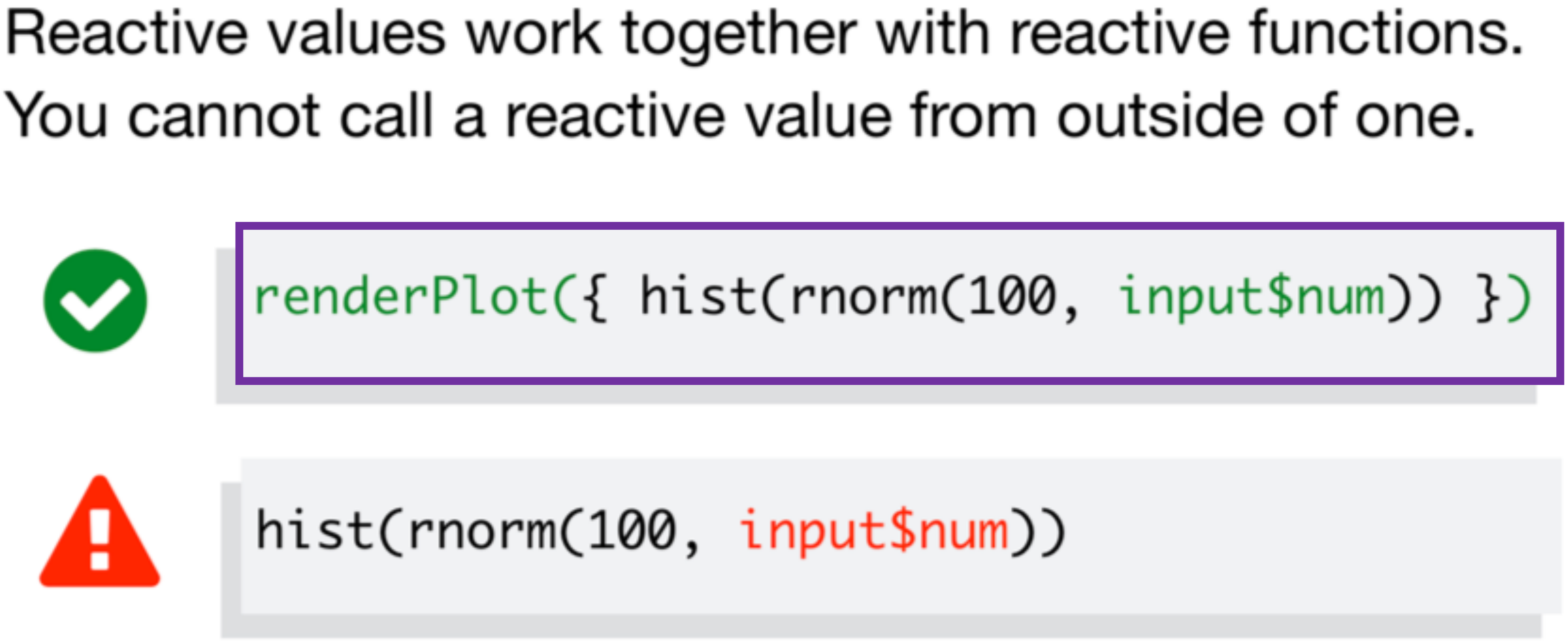


Choose a number

1 27 100

bit.ly/shiny-quickstart-2

A simple worked example



Reactive values work together with reactive functions. You cannot call a reactive value from outside of one.

✓

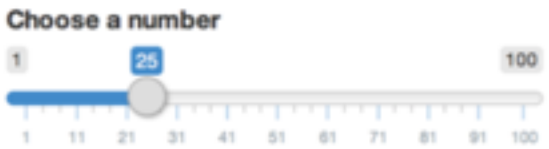
```
renderPlot({ hist(rnorm(100, input$num)) })
```

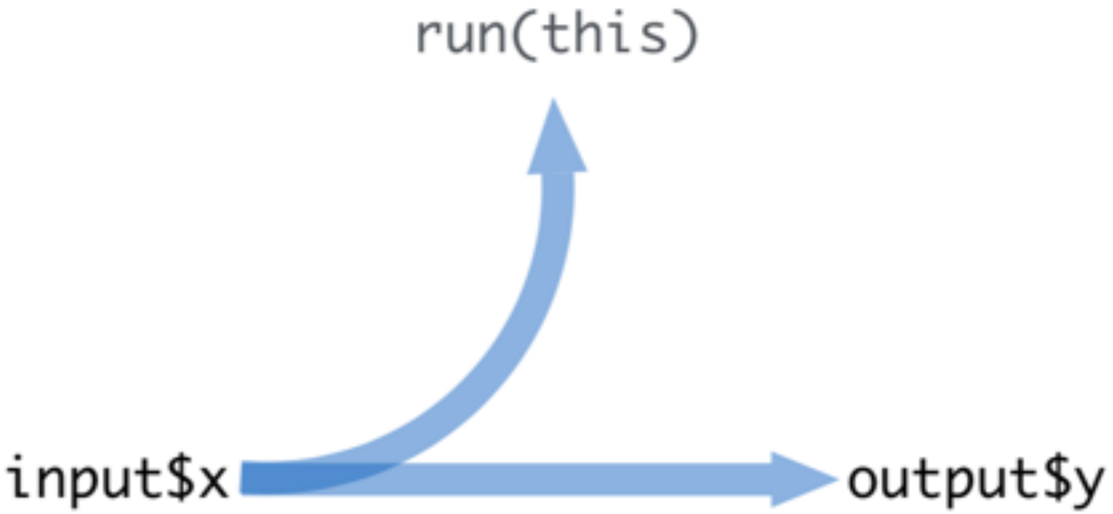
⚠

```
hist(rnorm(100, input$num))
```

A simple worked example

Just one example of reactivity in Shiny

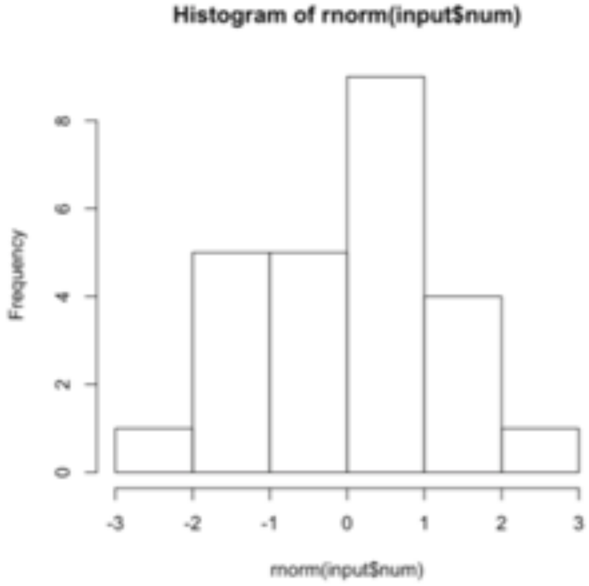




run(this)

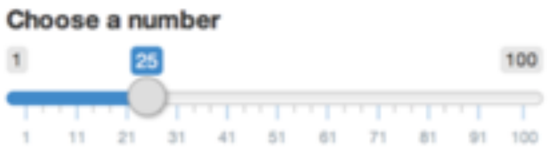
input$x

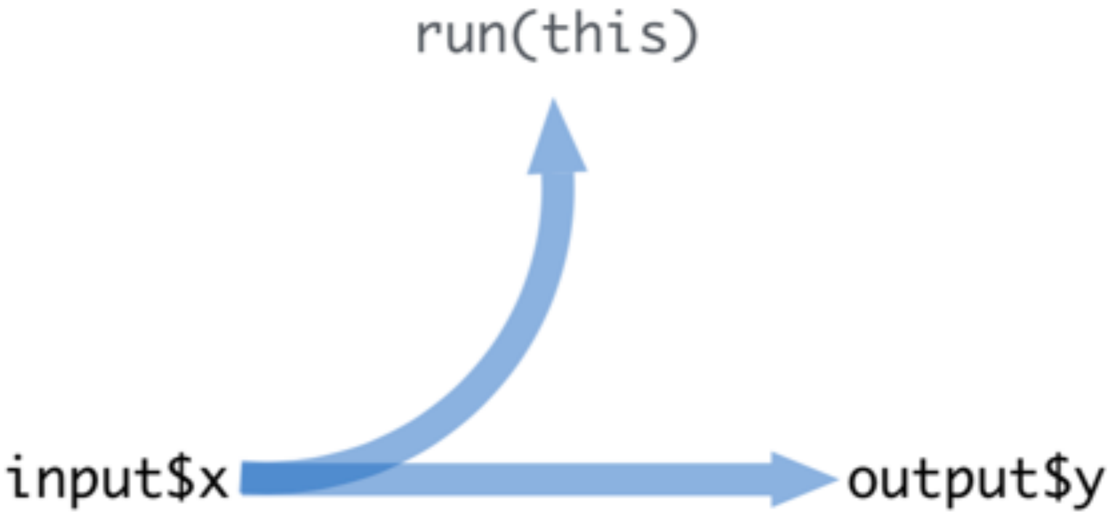
output$y



Reactivity in Shiny – a Very Brief Introduction

Just one example of reactivity in Shiny

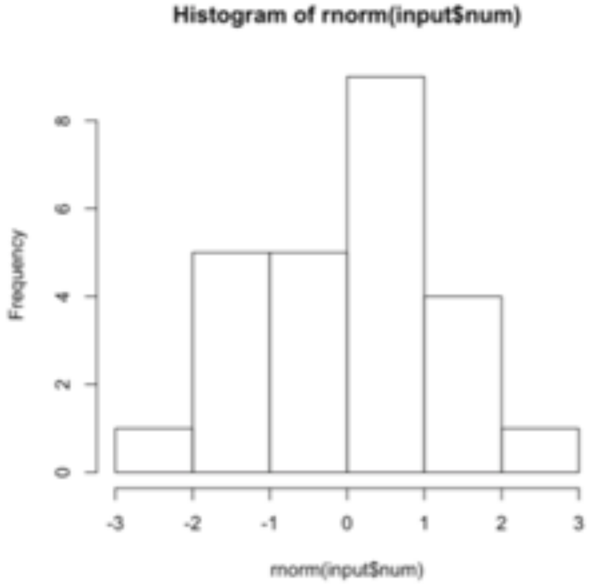




run(this)

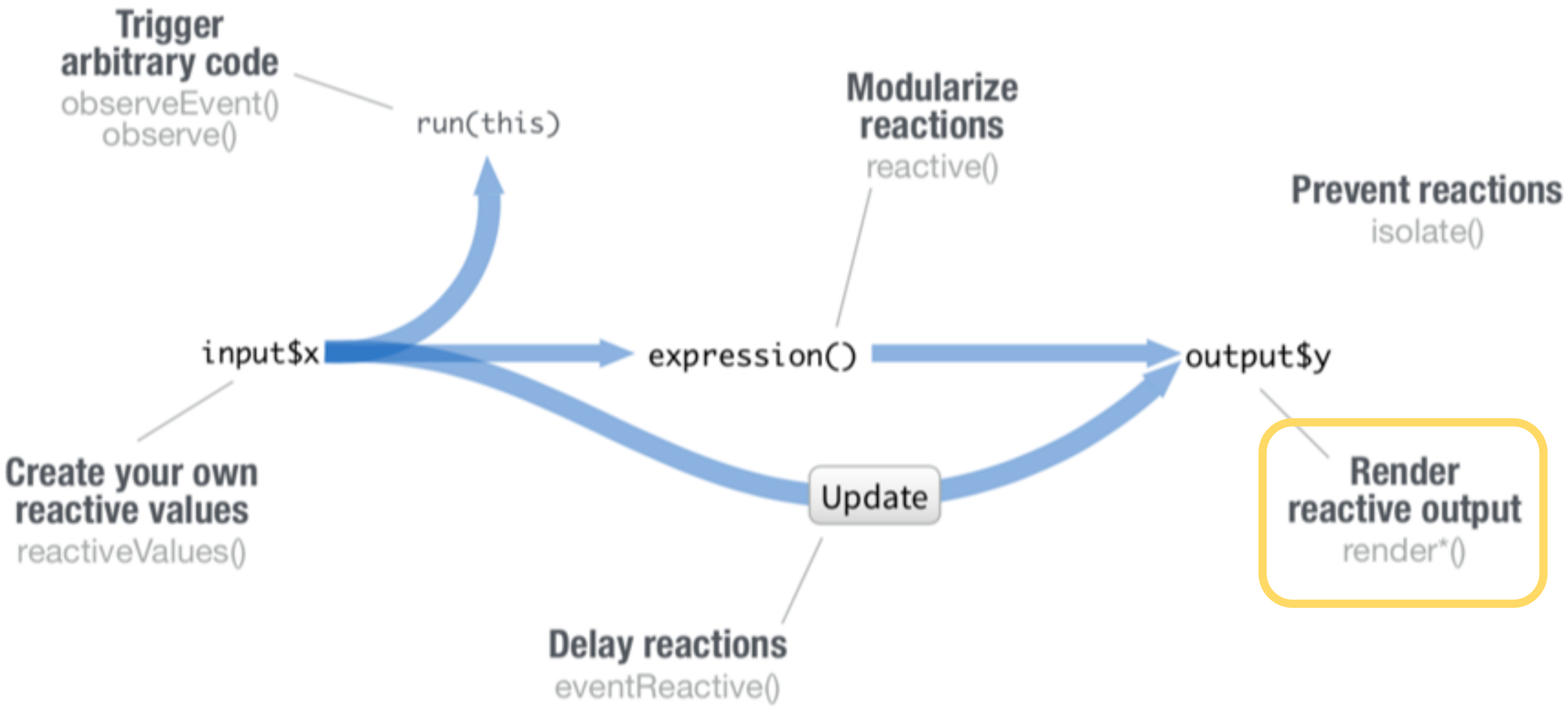
input$x

output$y



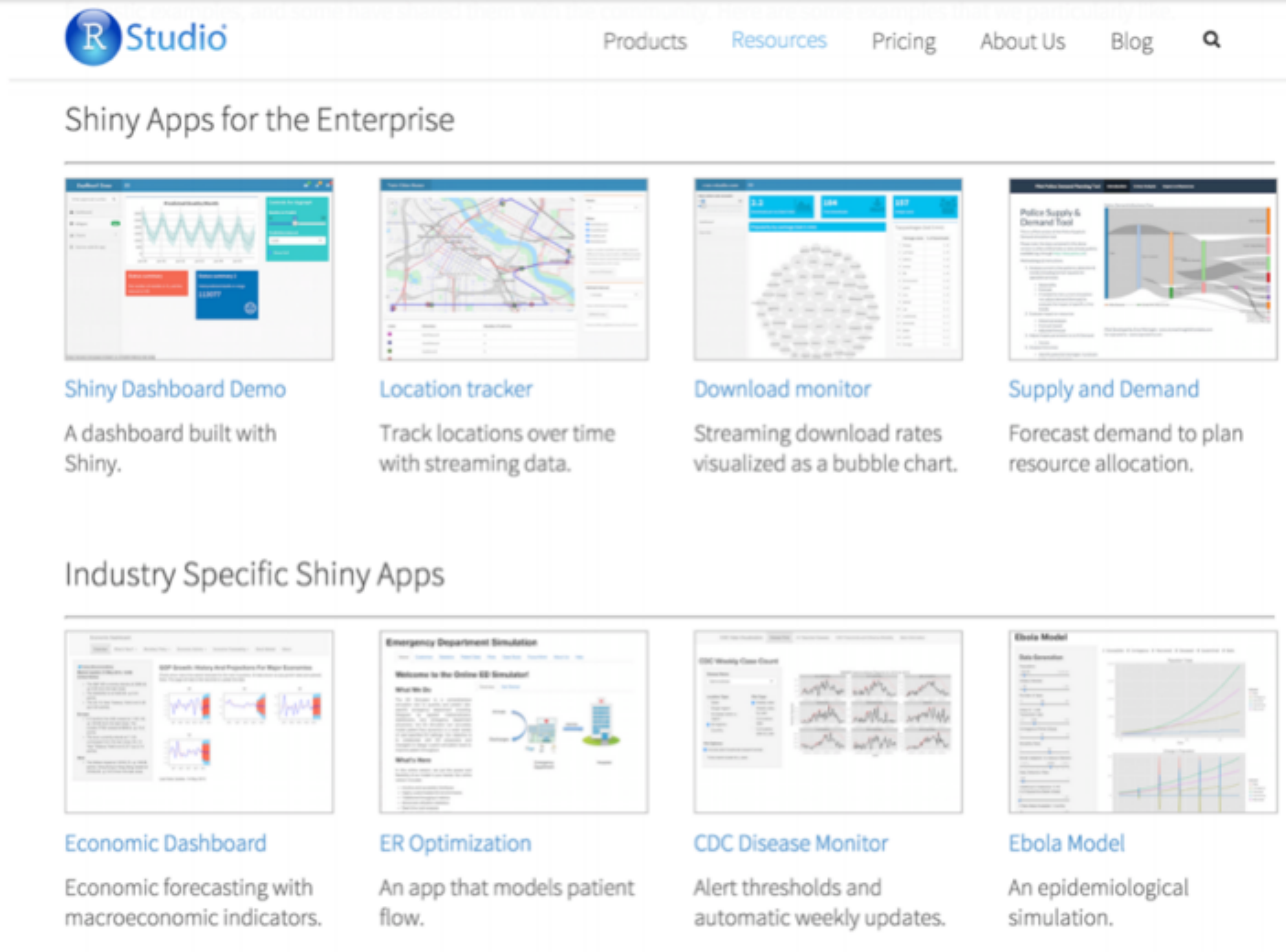
Reactivity in Shiny – a Very Brief Introduction

In fact it can be much more complex



See more online at the Shiny Gallery!

https://www.rstudio.com/products/shiny/shiny-user-showcase/



Shiny Apps for the Enterprise

Shiny Dashboard Demo
A dashboard built with Shiny.

Location tracker
Track locations over time with streaming data.

Download monitor
Streaming download rates visualized as a bubble chart.

Supply and Demand
Forecast demand to plan resource allocation.

Industry Specific Shiny Apps

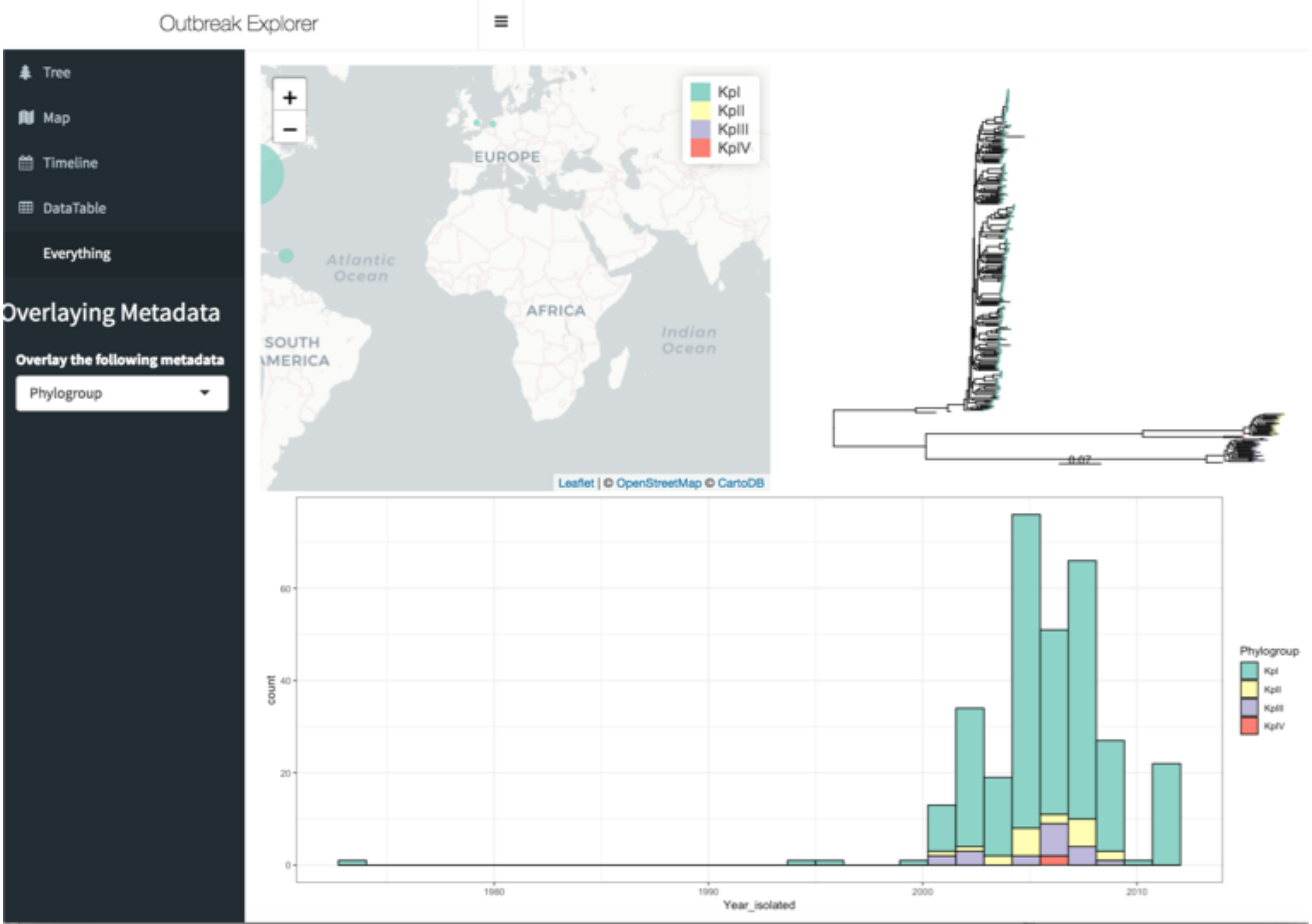
Economic Dashboard
Economic forecasting with macroeconomic indicators.

ER Optimization
An app that models patient flow.

CDC Disease Monitor
Alert thresholds and automatic weekly updates.

Ebola Model
An epidemiological simulation.

Now Let's for Ourselves!



Overlay the following metadata

Phylogroup ▾