

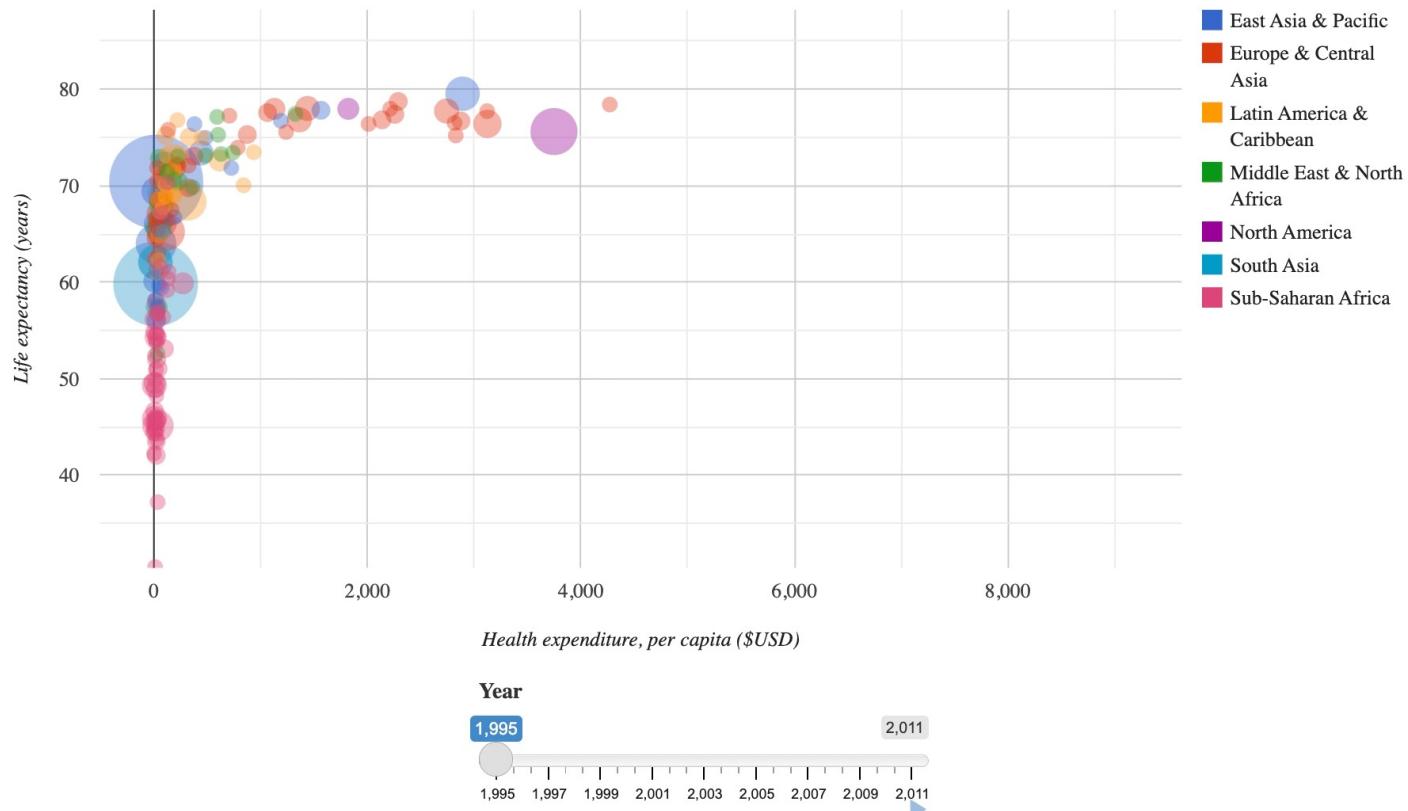
Section IIb

BASIC REACTIVITY

REACTIVITY

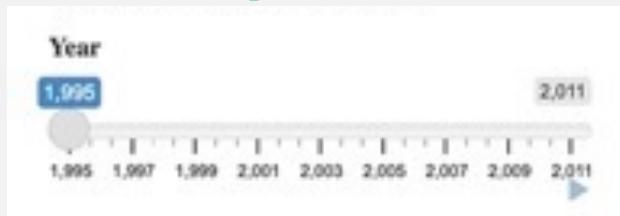
Google Charts demo

Health expenditure vs. life expectancy, 1995



HOW DOES IT WORKS?

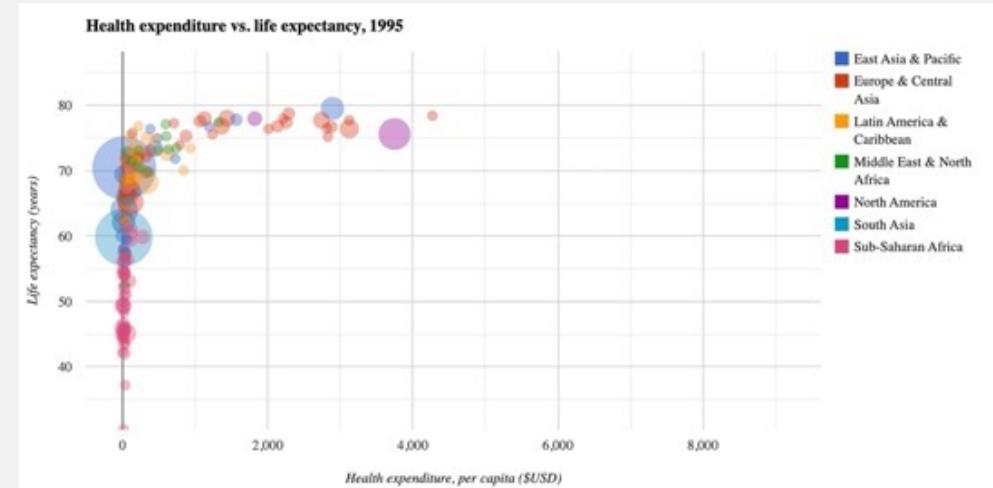
input\$x



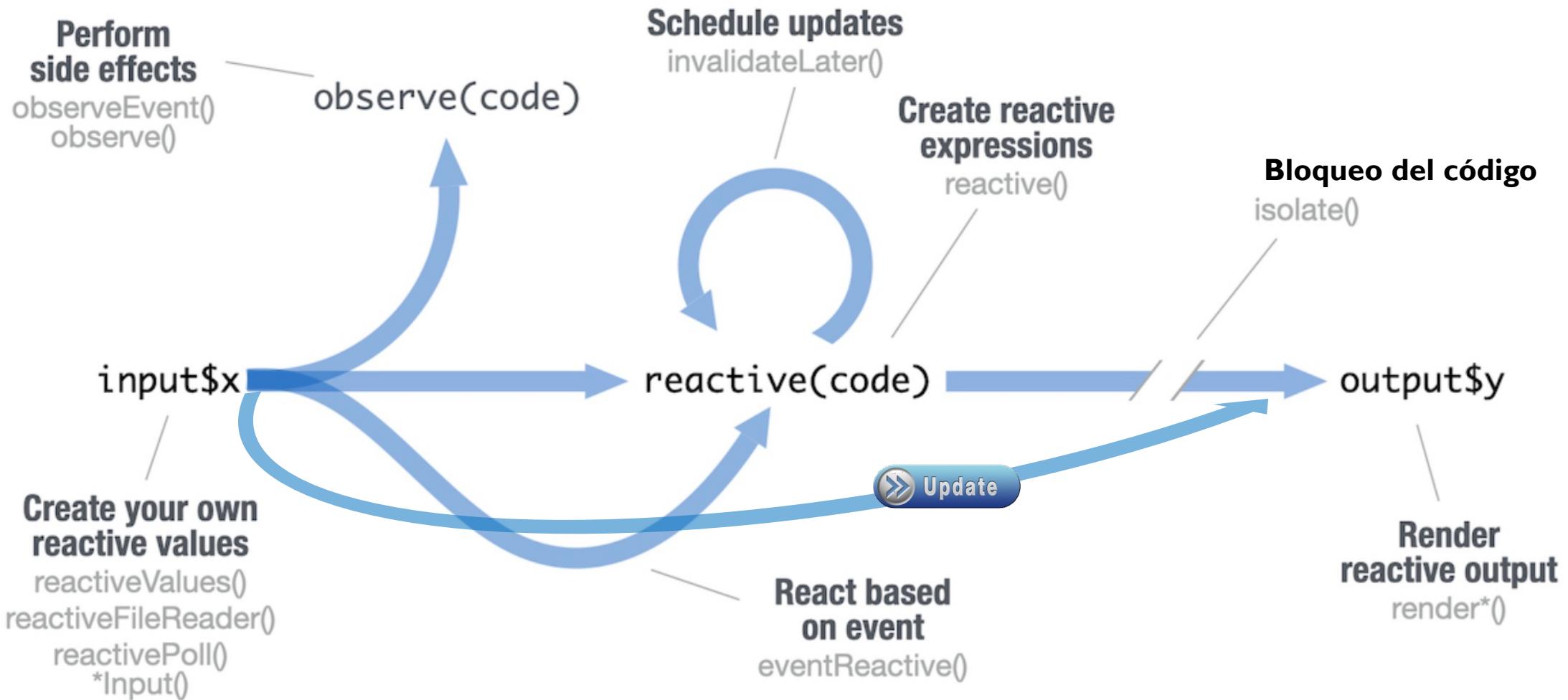
reactive ()*



output\$y



REACTIVE FUNCTIONS



RENDER FUNCTION ()*

Función	Objeto
renderDataTable ()	Interactive table
renderImage ()	Image
renderPlot ()	Plot
renderPrint ()	Block of code
renderTable ()	Table
renderText ()	Text
renderUI ()	Shiny UI element

DEFINING OUTPUTS TO SHOW IN THE UI

```
library(shiny)

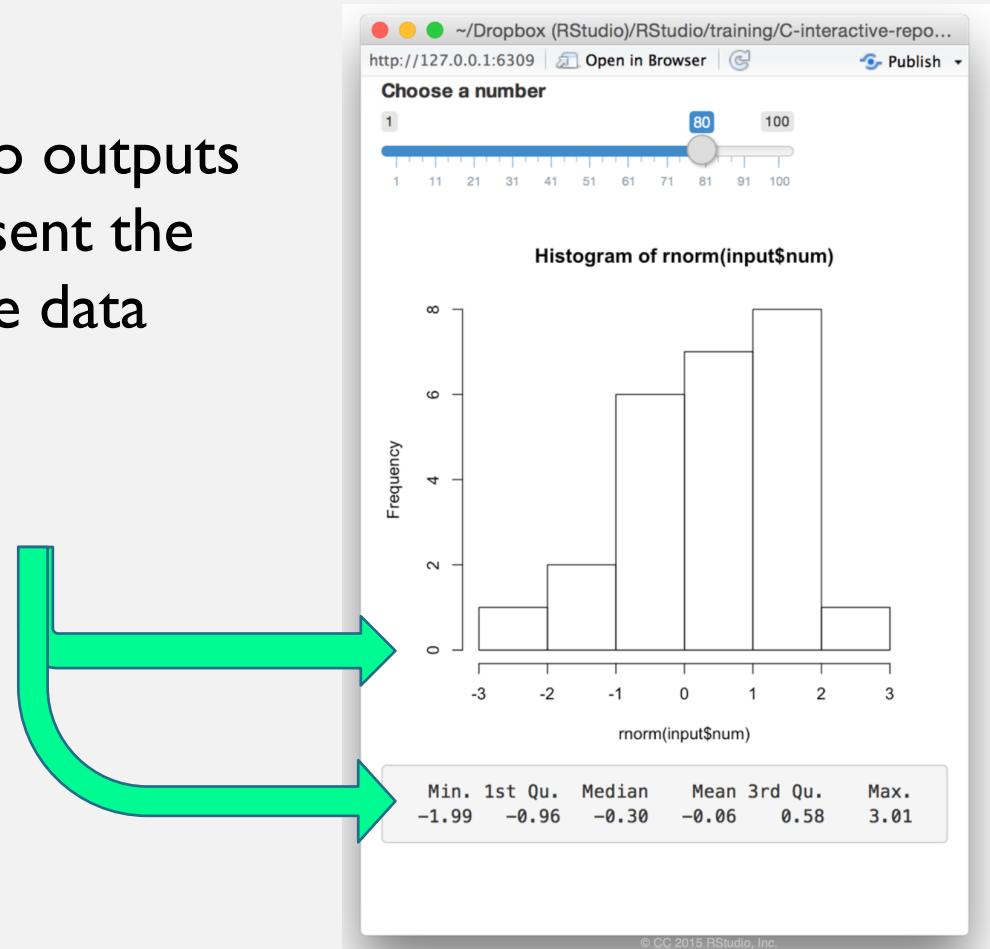
ui <- fluidPage(
  sliderInput(inputId = "num",
  label = "Choose a number",
  value = 25, min = 1, max = 100),
  plotOutput("hist")
)
server <- function(input, output) {
  output$hist <- renderPlot ({
    hist(rnorm(input$num))
  })
}
shinyApp(ui = ui, server = server)
```

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

TWO OUTPUTS, SAME REACTIVE OBJECT

```
library(shiny)
ui <- fluidPage(
  sliderInput(inputId = "num",
  label = "Choose a number",
  value = 25, min = 1, max = 100),
  plotOutput("hist"),
  verbatimTextOutput("stats")
)
server <- function(input, output) {
  output$hist <- renderPlot ({
    hist(rnorm(input$num))
  })
  output$stats <- renderPrint({
    summary(rnorm(input$num))
  })
}
shinyApp(ui = ui, server = server)
```

This two outputs
represent the
same data



DEFINING OUR REACTIVE OBJECTS

```
library(shiny)  
  
ui <- fluidPage(  
  sliderInput(inputId = "num",  
    label = "Choose a number",  
    value = 25, min = 1, max = 100),  
  plotOutput("hist"),  
  verbatimTextOutput("stats")  
)  
  
server <- function(input, output) {  
  data <- reactive({ rnorm(input$num)}  
  
    output$hist <- renderPlot ({  
      hist(data())  
    })  
    output$stats <- renderPrint({  
      summary(data())  
    })  
}  
  
shinyApp(ui = ui, server = server)
```

data <- reactive({rnorm(input\$num)})

We use **data ()** for our histogram function

**output\$hist <- renderPlot ({
 hist(data())
})**

ACTION BUTTONS



Input function

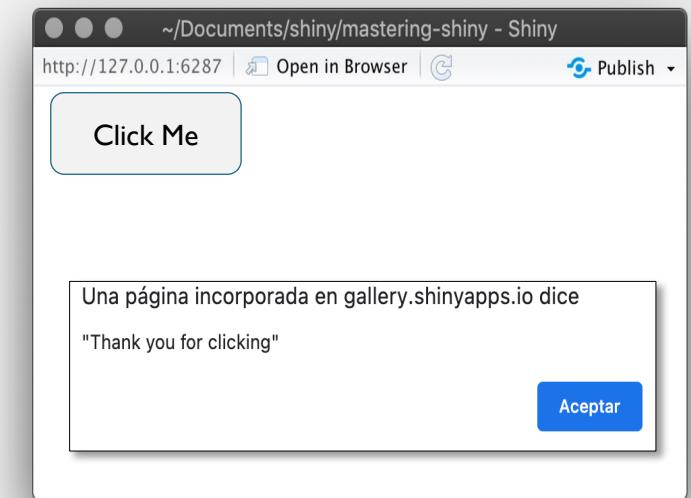
input ID

Label to show

```
actionButton(inputId = "go", label = "Click Me")
```

ACTION BUTTONS IN THE SERVER

```
library(shiny)  
ui <- fluidPage(  
  actionButton(inputId = "clicks",  
               label = "Click me")  
)  
  
server <- function(input, output) {  
  observeEvent(input$clicks, {  
    session$sendCustomMessage(type = 'testmessage',  
                               message = 'Thank you for clicking')  
  })  
}  
  
shinyApp(ui = ui, server = server)
```



ACTION BUTTONS HELP CONTROLLING REACTIVITY

```
library(shiny)  
  
ui <- fluidPage(  
  sliderInput(inputId = "num",  
    label = "Choose a number",  
    value = 25, min = 1, max = 100),  
  actionButton(inputId = "go", label = "Update"),  
  plotOutput("hist")  
)  
  
server <- function(input, output) {  
  data <- eventReactive(input$go, {  
    rnorm(input$num)  
  })  
  output$hist <- renderPlot({  
    hist(data())  
  })  
}  
  
shinyApp(ui = ui, server = server)
```

