R Shiny - Part II

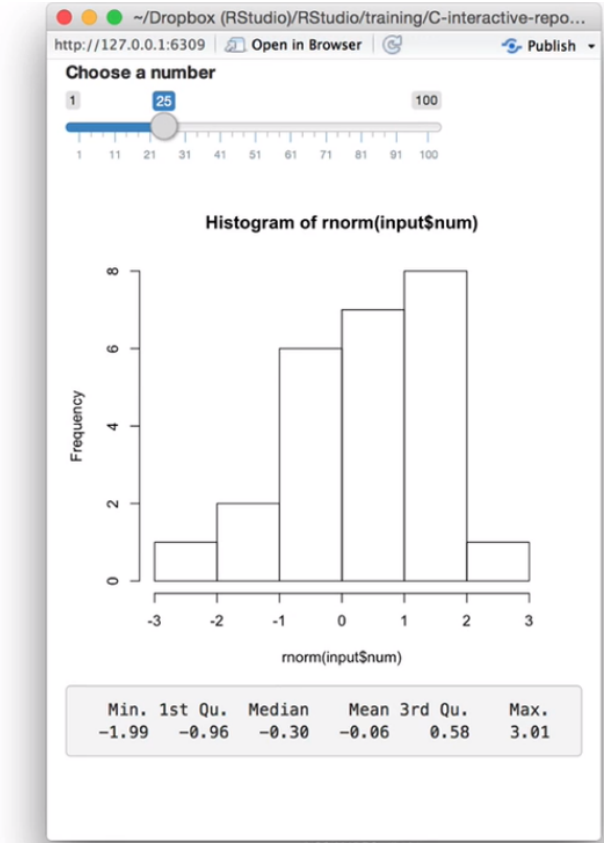
Workshop on Plotting in R

Lokesh Mano • 22-Jul-2021
NBIS, SciLifeLab

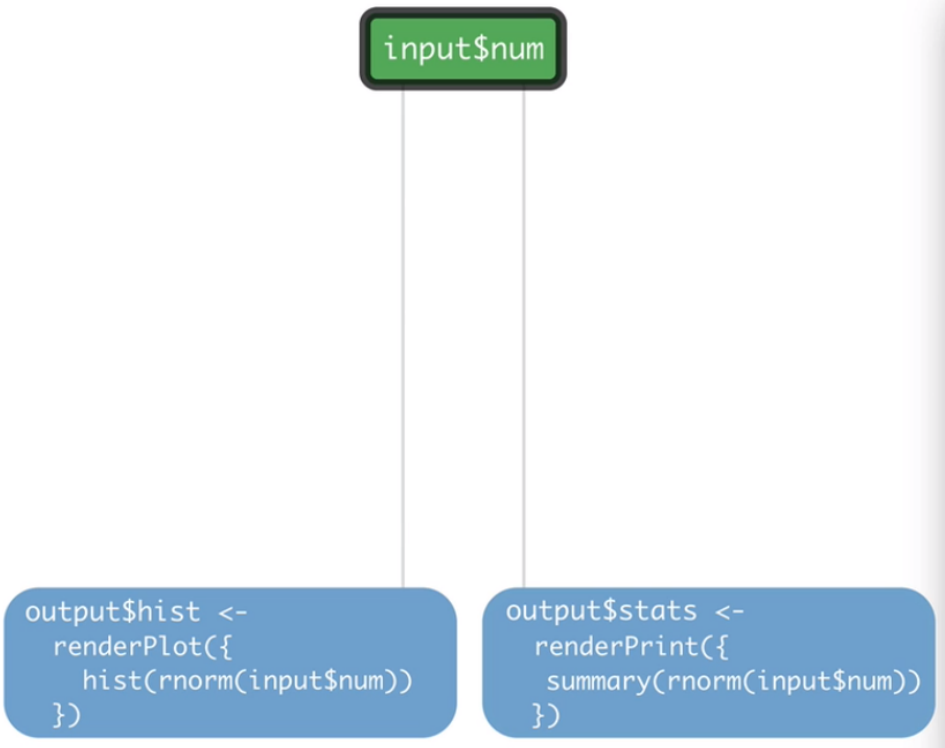
Contents

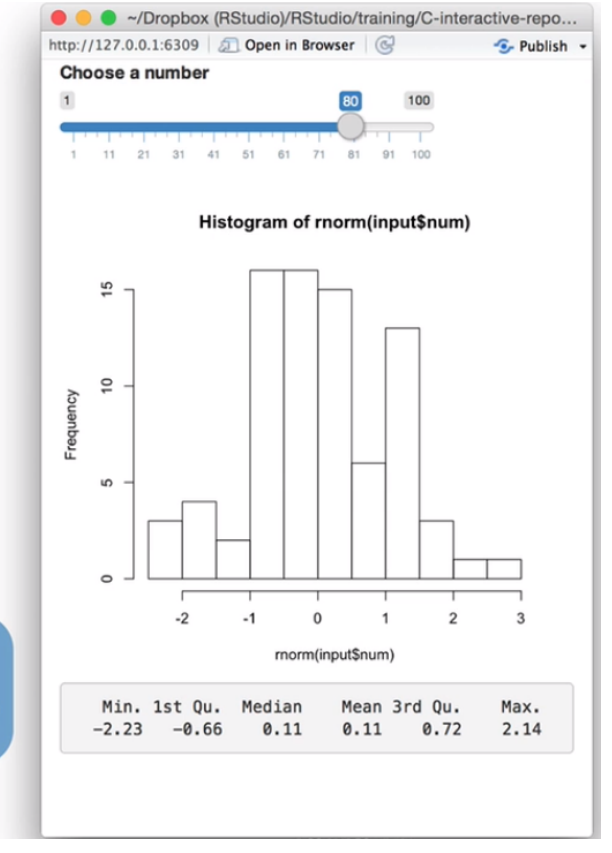
Reactivity

```
# 02-two-outputs

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)
```

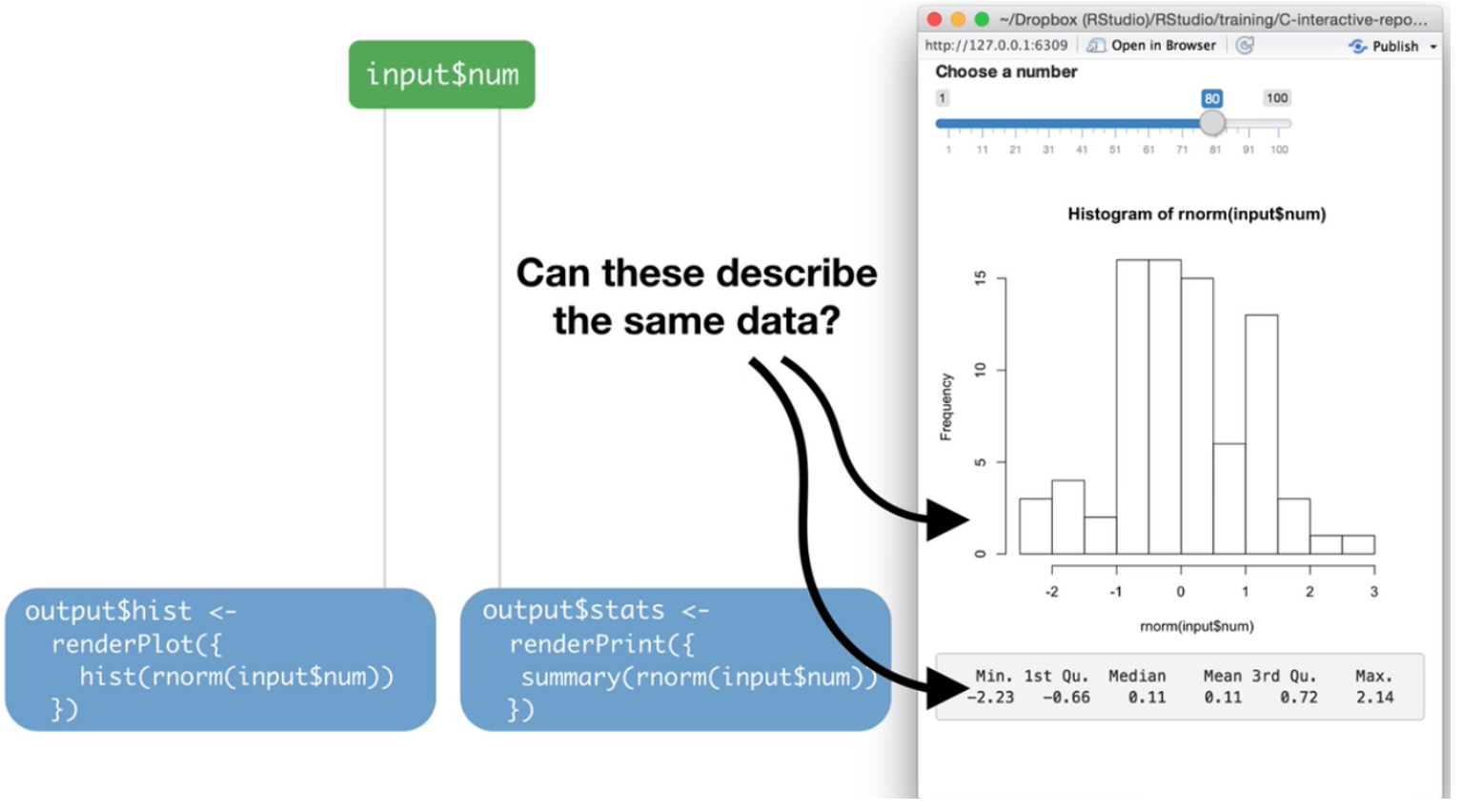


Reactivity

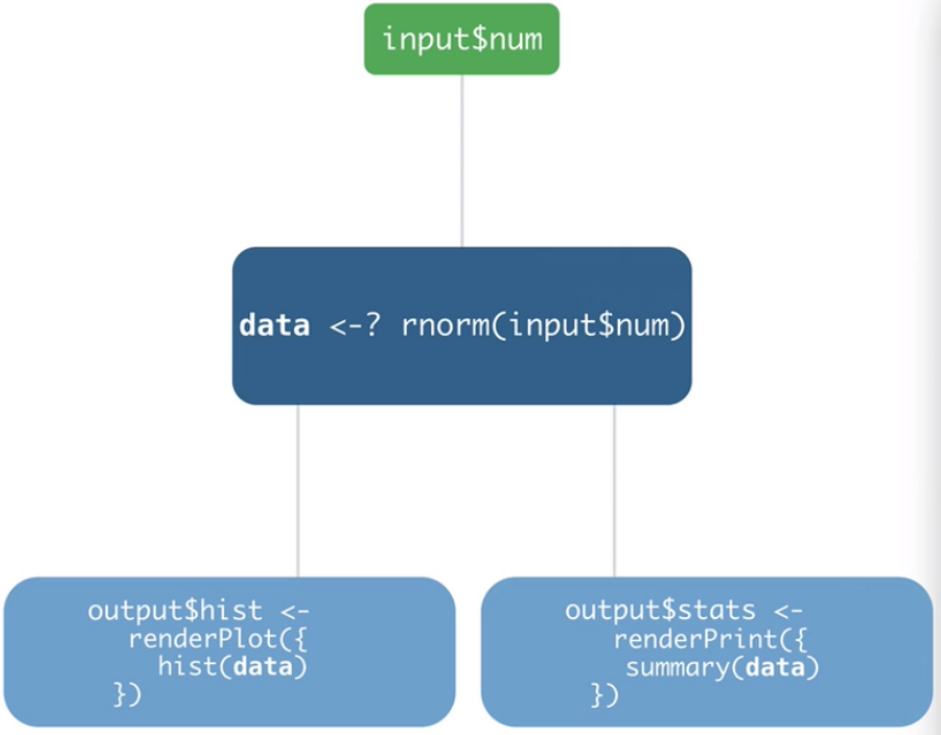

input$num
output$hist <-
renderPlot({
hist(rnorm(input$num))
})
output$stats <-
renderPrint({
summary(rnorm(input$num))
})

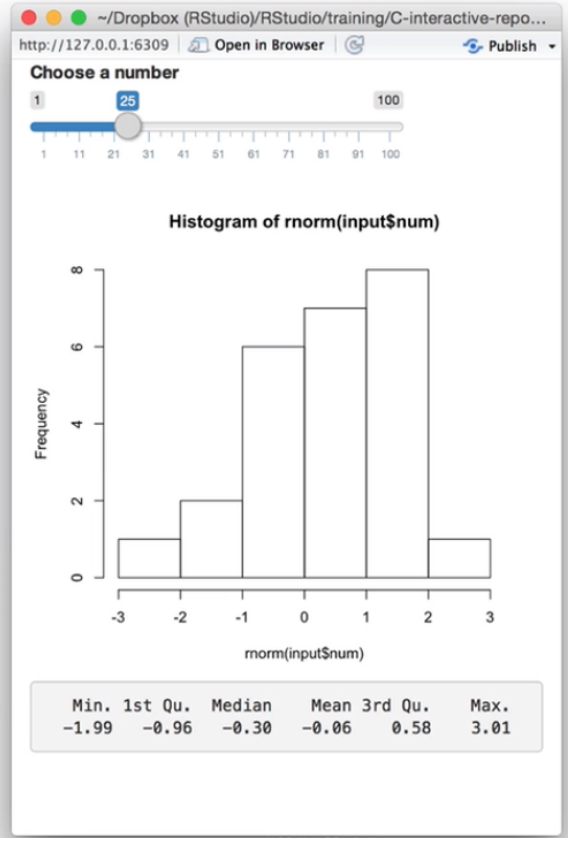

~/Dropbox (RStudio)/RStudio/training/C-interactive-repo...
http://127.0.0.1:6309
Open in Browser
Publish
Choose a number
1
80
100
1 11 21 31 41 51 61 71 81 91 100
Histogram of rnorm(input$num)
Frequency
0 5 10 15
-2 -1 0 1 2 3
rnorm(input$num)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-2.23 -0.66 0.11 0.11 0.72 2.14

Reactivity


input$num
Can these describe the same data?
output$hist <-
renderPlot({
hist(rnorm(input$num))
})
output$stats <-
renderPrint({
summary(rnorm(input$num))
})
~/Dropbox (RStudio)/RStudio/training/C-interactive-repo...
http://127.0.0.1:6309
Open in Browser
Publish
Choose a number
1
80
100
Histogram of rnorm(input$num)
Frequency
rnorm(input$num)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-2.23 -0.66 0.11 0.11 0.72 2.14

Reactivity


input$num
data <-? rnorm(input$num)
output$hist <-
renderPlot({
hist(data)
})
output$stats <-
renderPrint({
summary(data)
})


~/Dropbox (RStudio)/RStudio/training/C-interactive-repo...
http://127.0.0.1:6309
Open in Browser
Publish
Choose a number
1
25
100
Histogram of rnorm(input$num)
Frequency
rnorm(input$num)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-1.99 -0.96 -0.30 -0.06 0.58 3.01

Reactivity

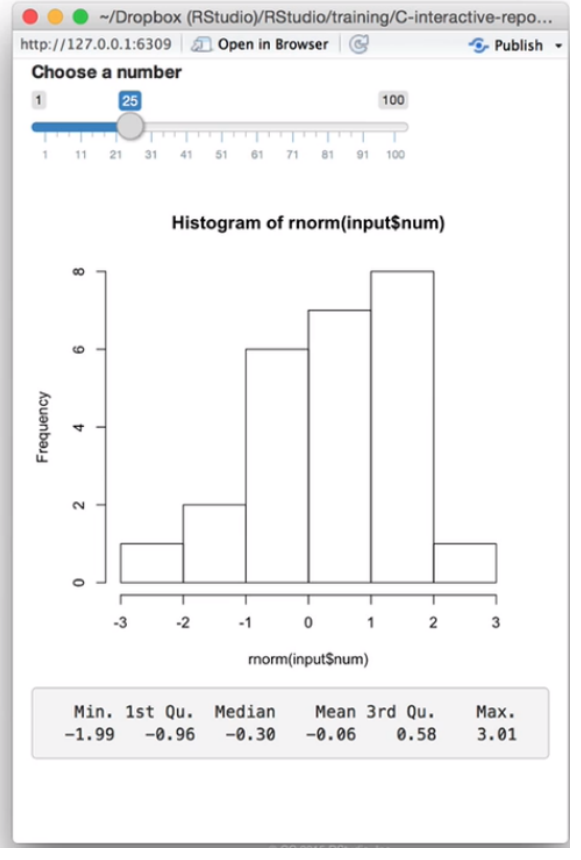
reactive()

Builds a reactive object (reactive expression)

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

Reactivity

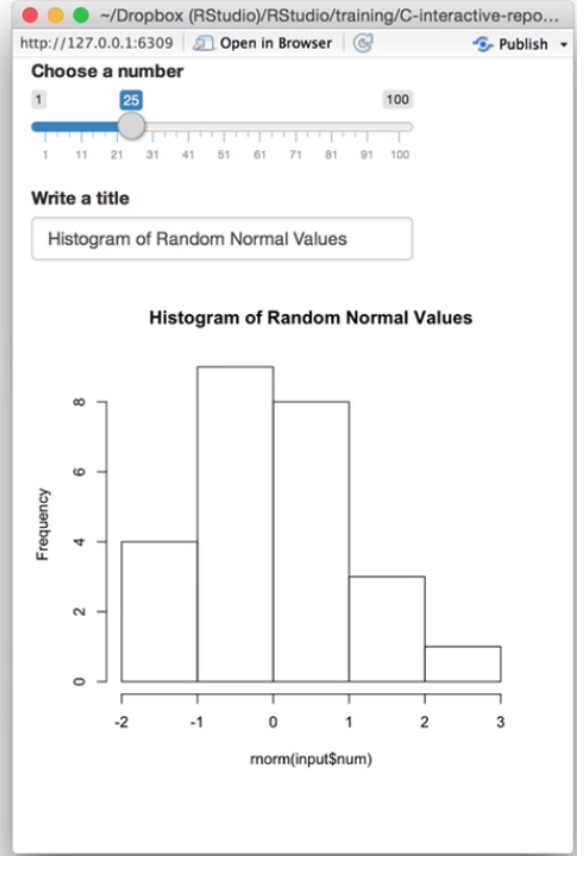
```
# 03-reactive

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)
```



Isolate reactivity

- Reactivity can be controlled.* You will notice that as soon as you try to change the title the histogram will update with new values

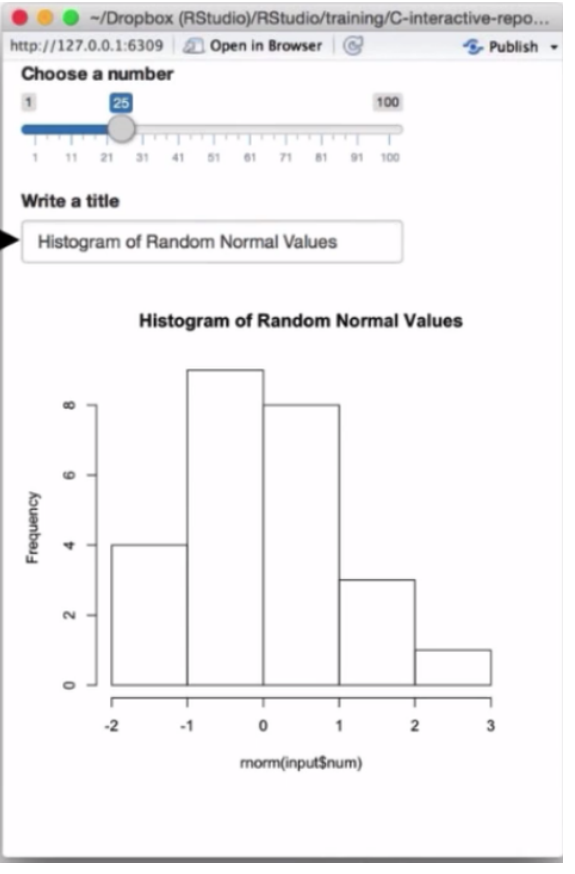
```
# 01-two-inputs

library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  textInput(inputId = "title",
    label = "Write a title",
    value = "Histogram of Random Normal Values"),
  plotOutput("hist")
)

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

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



Isolate reactivity

```
# 01-two-inputs

library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  textInput(inputId = "title",
    label = "Write a title",
    value = "Histogram of Random Normal Values"),
  plotOutput("hist")
)

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

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

Can we prevent the title field from updating the plot?



Isolate reactivity

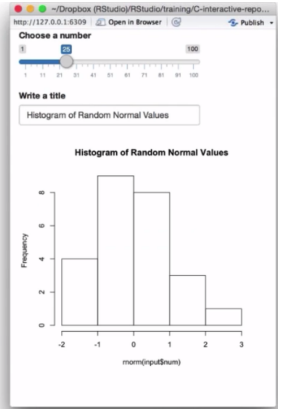
isolate()

Returns the result as a non-reactive value

```
isolate({ rnorm(input$num) })
```

Isolate reactivity

```
# 04-isolate

library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  textInput(inputId = "title",
    label = "Write a title",
    value = "Histogram of Random Normal Values"),
  plotOutput("hist")
)

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

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



Thank you. Questions?

Slide courtesy: Roy Francis (NBIS, RaukR2021)

R version 4.1.0 (2021-05-18)

Platform: x86_64-pc-linux-gnu (64-bit)

OS: Ubuntu 18.04.5 LTS

Built on : 22-Jul-2021 at 14:36:26