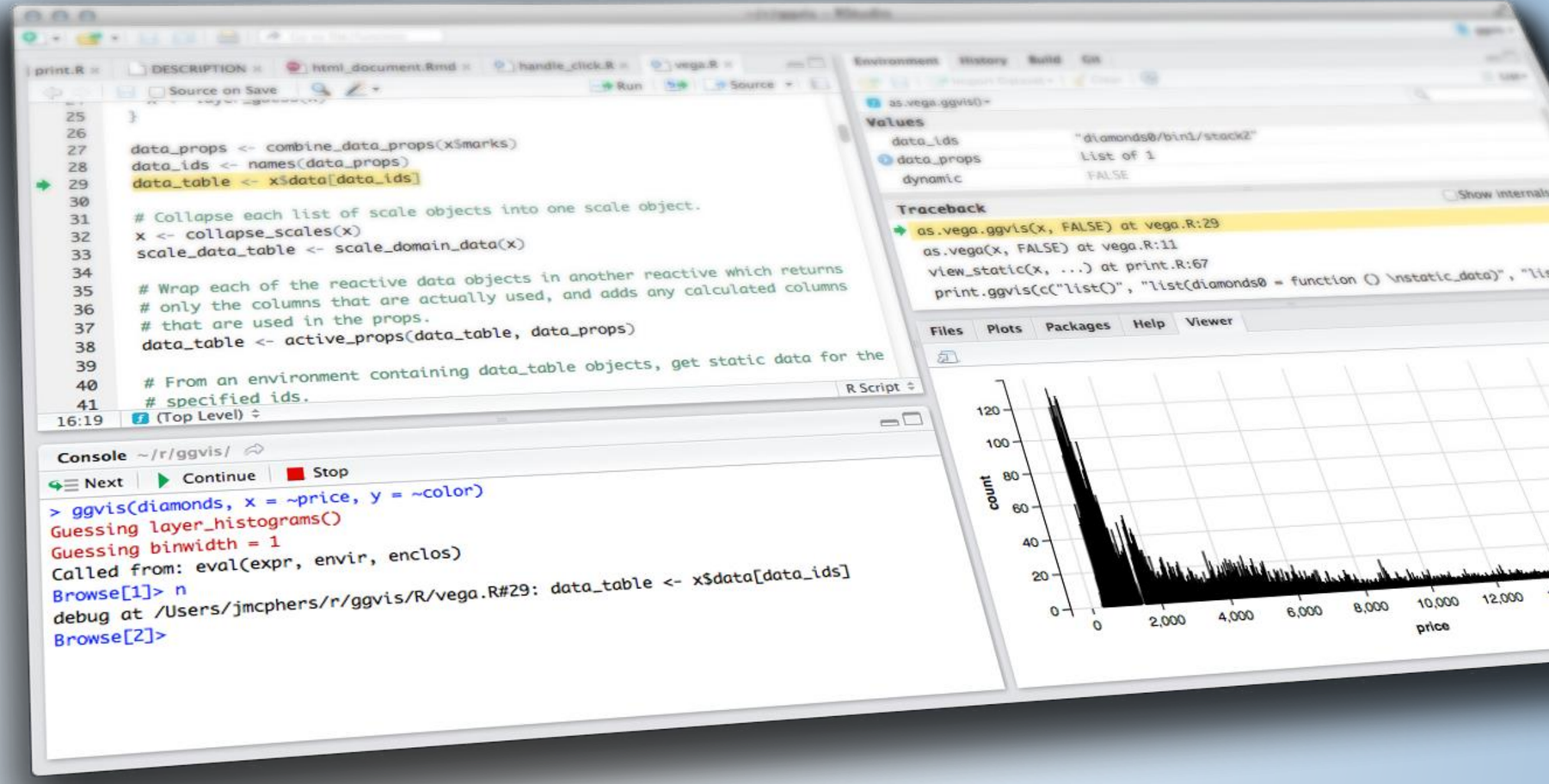


BOOKMARKING

Shiny from RStudio™

OUTLINE

- Bookmarking
  - How to bookmark
  - Bookmarking options
  - Customizing bookmarks

Motivation

MOTIVATION

- Capture the results you're currently seeing in a Shiny app, by snapshotting inputs and state
- Creates a URL that can be shared with others, or bookmarked for your own future use
- Applications:
  - Collaboration across researchers: Biostatistician creates an app, passes on to the biologist to explore, biologist wants to communicate specific findings back to the biostatistician
  - Sharing results with a coworker/manager: Instead of sending someone to your app and then asking them to click this, check that, choose this, etc.
  - Education:
    - If using a Shiny app to teach concepts, students might be asked to play around with a distribution by tweaking inputs, and submit the output
    - Or an instructor might want to provide multiple scenarios in an app for students to explore

How to bookmark

A VERY SIMPLE APP

```
ui <- fluidPage(
    textInput("txt", "Enter text"),
    checkboxInput("caps", "Capitalize"),
    verbatimTextOutput("out")
  )

server <- function(input, output, session) {
  output$out <- renderText({
    ifelse(input$caps, toupper(input$txt), input$txt)
  })
}

shinyApp(ui, server)
```

ENABLING BOOKMARKING

```
ui <- function(request) {
  fluidPage(
    textInput("txt", "Enter text"),
    checkboxInput("caps", "Capitalize"),
    verbatimTextOutput("out"),
    bookmarkButton()
  )
}

server <- function(input, output, session) {
  output$out <- renderText({
    ifelse(input$caps, toupper(input$txt), input$txt)
  })
}

shinyApp(ui, server, enableBookmarking = "url")
```

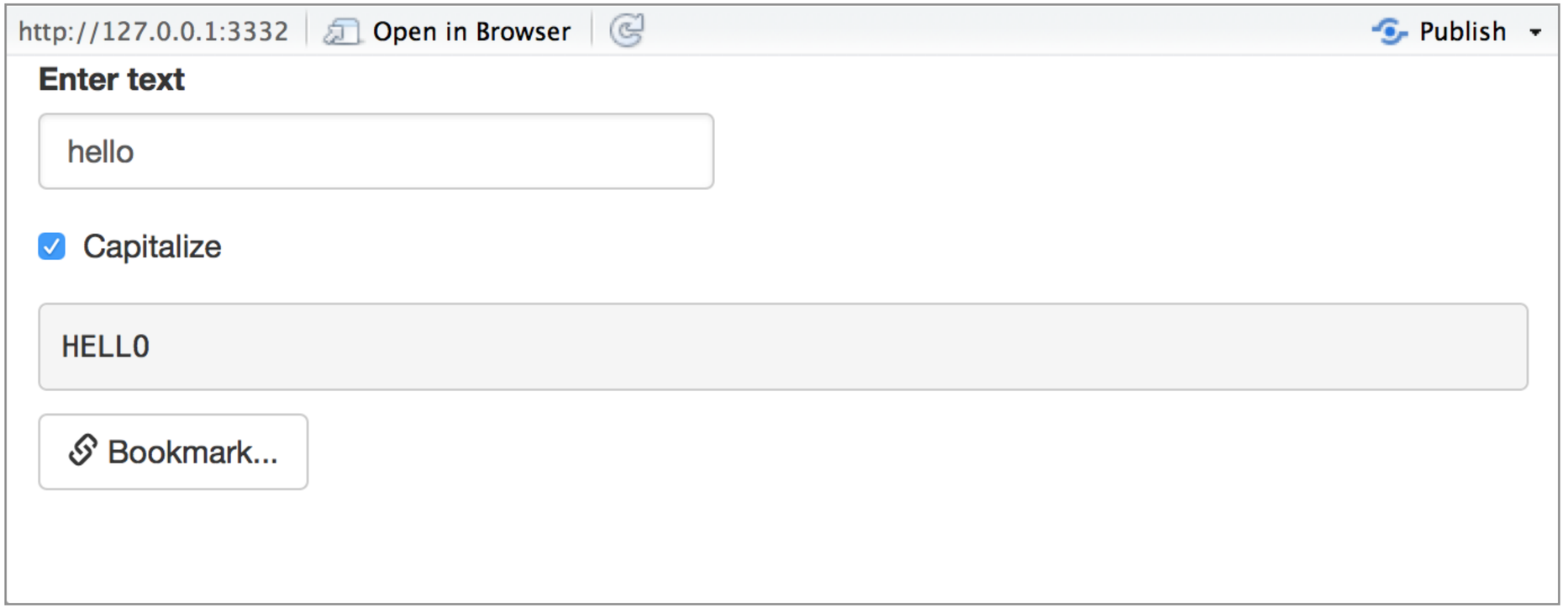
UI portion must be a function that takes one argument

A button for bookmarking

There must be a call to enableBookmarking()

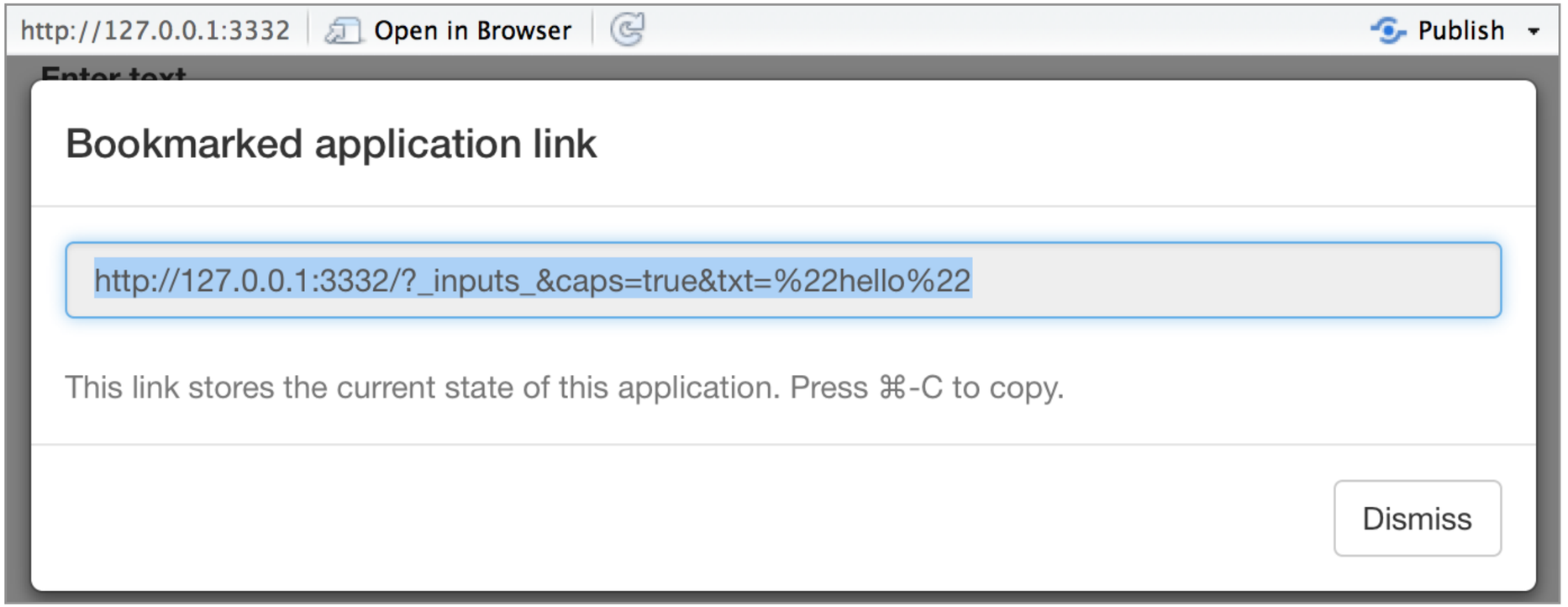
bookmark_01.R

DEMO



bookmark_01.R

DEMO



bookmark_01.R

DEMO

HOW BOOKMARKING WORKS

- Bookmarked state automatically saves the input values (except passwords) such that when the application is restored using that state, the inputs are seeded with the saved values
- File inputs are saved only when state is saved to server (not with URL encoding)

Bookmarking options

TYPES OF BOOKMARKING

URL-encoded	Saved-to-server
No state on server	State saved on server
URLs may be long & values revealed in URL	Always a short URL
URL length limits data (~2K characters on some browsers)	No limits on data
Can’t store uploaded files	Can store uploaded files
Best for simple apps without too much state to serialize	Appropriate for large amounts of state, incl files and directories if necessary

ENABLING BOOKMARKING

```
ui <- function(request) {
  fluidPage(
    textInput("txt", "Enter text"),
    checkboxInput("caps", "Capitalize"),
    verbatimTextOutput("out"),
    bookmarkButton()
  )
}

server <- function(input, output, session) {
  output$out <- renderText({
    ifelse(input$caps, toupper(input$txt), input$txt)
  })
}

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

Only change for saving to server

DEPLOYED EXAMPLES

URL-encoded	Saved-to-server
https://gallery.shinyapps.io/113-bookmarking-url/	https://gallery.shinyapps.io/bookmark-saved/?_state_id_=d80625dc681e913a
https://gallery.shinyapps.io/113-bookmarking-url/?_inputs_&n=200	

Customizing bookmarks

EXCLUDING INPUTS

- Inputs that should not be bookmarked can be excluded with setBookmarkExclude() in the server function, which takes in a vector containing the names of the inputs

```
server <- function(input, output, session) {
  setBookmarkExclude(c("x", "y"))
}
```

MULTIPLE TABS

- It is possible to bookmark which tab in a tabset is active
  - This requires providing IDs for `tabsetPanel()`, `navbarPage()`, or `navlistPanel()`
- If you have multiple tabs, you likely want the bookmark button to show up on each tab
  - This also requires providing IDs for `bookmarkButton()`, and also excluding the bookmarking of the bookmark buttons themselves

bookmark_02.R

Two tabs with checkboxes, and bookmark button on each tab

DEMO

```
server <- function(input, output, session) {

  # Need to exclude the buttons from themselves being bookmarked
  setBookmarkExclude(c("bookmark1", "bookmark2"))

  # Trigger bookmarking with either button
  observeEvent(input$bookmark1, {
    session$doBookmark()
  })
  observeEvent(input$bookmark2, {
    session$doBookmark()
  })
}
```

To trigger bookmarking from each button, use **observeEvent()** for each button that calls session**$doBookmark()**.

INPUT DEPENDENCE

- **Fully input-dependent apps:** The state of the outputs at time *t* is fully determined by the state of the inputs at time *t*
  - Bookmarking should “just work”
- **Partly input-dependent apps:** The state of the outputs at time *t* is only partly determined by the state of the inputs at time *t* — other things may influence it, including inputs at previous times, or external data
  - These apps are not fully reactive
  - Bookmarking requires additional tools to save & restore desired state

bookmark_03.R

DEMO

Displays the sum of all previous slider values added

```
ui <- fluidPage(
    sidebarPanel(
      sliderInput("n", "Value to add", min = 0, max = 100, value =
50),
      actionButton("add", "Add"),
   ),
    mainPanel(
      h4("Sum:", textOutput("sum"))
    )
  )

server <- function(input, output, session) {
  vals <- reactiveValues(sum = 0)

  observeEvent(input$add, {
    vals$sum <- vals$sum + input$n
  })
  output$sum <- renderText({
    vals$sum
  })
}

shinyApp(ui, server, enableBookmarking = "url")
```

State of output not fully determined by state of inputs since previous input values matter as well

Each time “Add” button is clicked, add input$n to vals$sum, then render text of the sum value

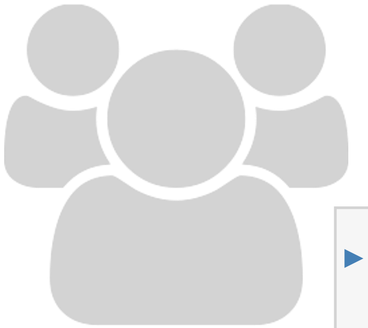
DEMO

bookmark_04.R

```
ui <- function(request) {
  fluidPage(
    sidebarPanel(
      sliderInput("n", "Value to add", min = 0, max = 100, value = 50),
      actionButton("add", "Add"),
    ),
    mainPanel(
      h4("Sum:", textOutput("sum"))
    )
    bookmarkButton()
  )
}

server <- function(input, output, session) {
  vals <- reactiveValues(sum = 0)

  onBookmark(function(state) {
    state$values$currentSum <- vals$sum
  })
  onRestore(function(state) {
    vals$sum <- state$values$currentSum
  })

  observeEvent(input$add, {
    vals$sum <- vals$sum + input$n
  })
  output$sum <- renderText({
    vals$sum
  })
}

shinyApp(ui, server, enableBookmarking = "url")
```

When app is bookmarked, save vals$sum in the bookmark state values

When app is restored, retrieve vals$sum from the bookmark state values



EXERCISE

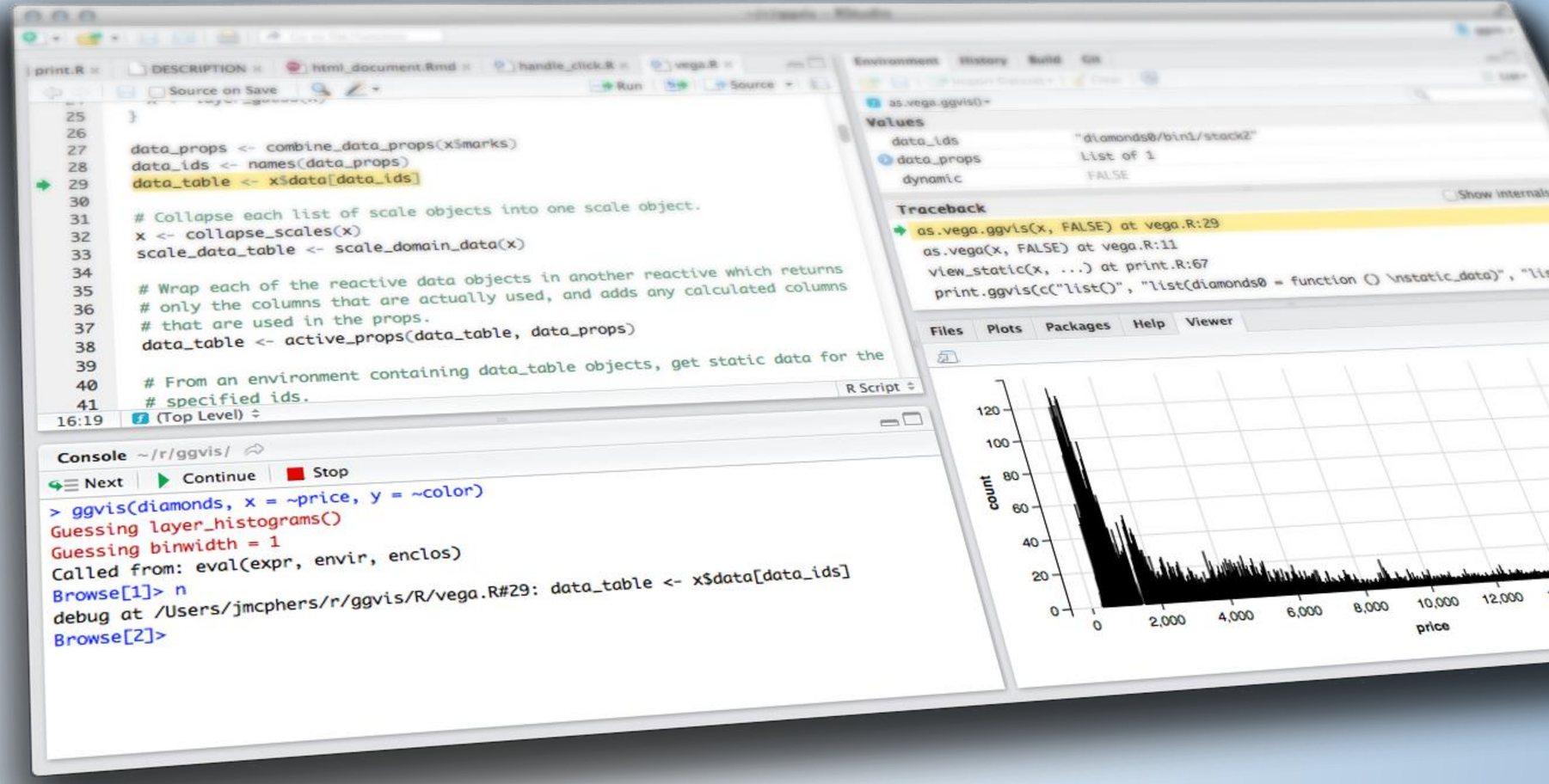
- Run bookmark_04.R
- There is a bug, what is it?
- Can you fix it?

3m 00s

SOLUTION

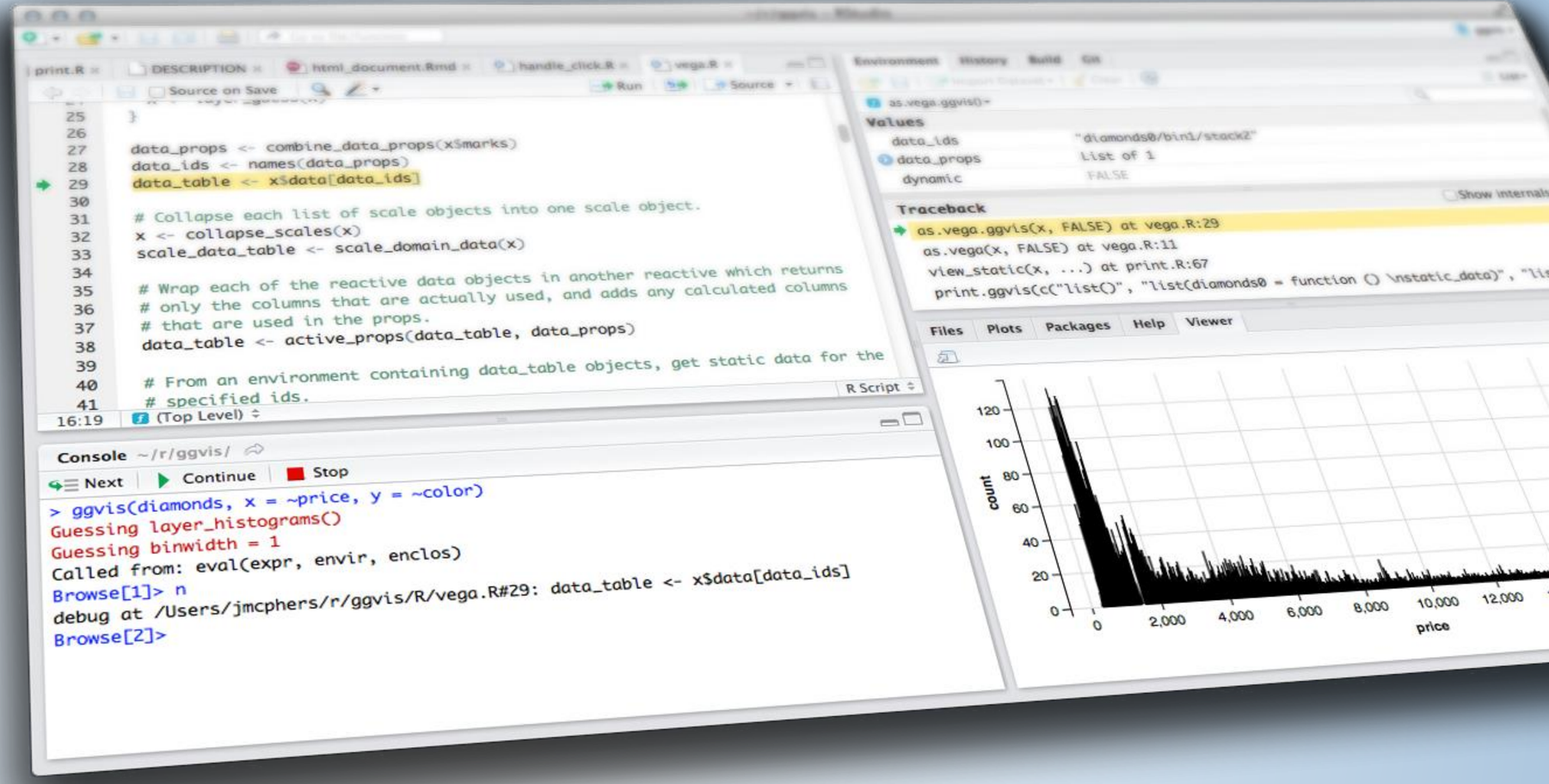
Solution to the previous exercise

bookmark_05.R



BOOKMARKING

Shiny from RStudio™



FINAL PROJECT OFFICE HRS

Shiny from RStudio™