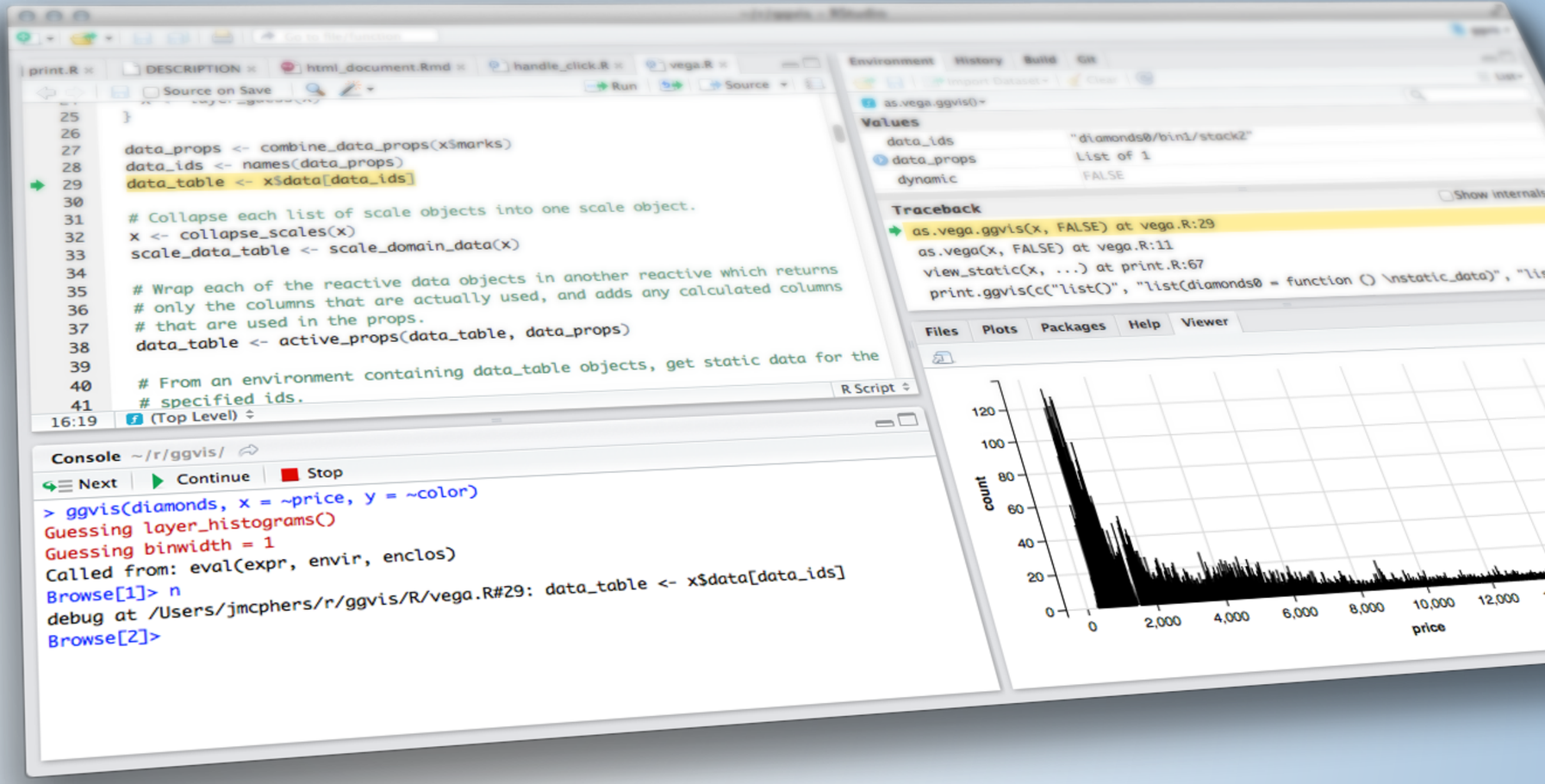


BOOKMARKABLE STATE

Shiny from RStudio™

OUTLINE

- Motivation
- Bookmarking options
- How to bookmark
- Advanced bookmarking

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:
  - 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
  - Collaboration across researchers: Biostatistician creates an app, passes on to the biologist to explore, biologist wants to communicate specific findings back to the biostatistician

Bookmarking options

TYPES OF BOOKMARKING

- **Encoding the state in a URL:**
  - Entire state of the application is contained in the URL's query string
  - Ex: https://gallery.shinyapps.io/113-bookmarking-url/ and https://gallery.shinyapps.io/113-bookmarking-url/?_inputs_&n=200
  - Best for simple apps without too much state to serialize
  - Values are revealed directly in the URL, this may not be desirable
- **Saving the state to the server:**
  - The state is saved to the server; the URL will contain an ID for the state
  - Ex: https://gallery.shinyapps.io/bookmark-saved/?_state_id_=d80625dc681e913a
  - Appropriate for large amounts of state (including files and directories if necessary)
  - **Hosting environment must provide support**

ENVIRONMENT SUPPORT

	Shiny Server Open Source	Shiny Server Pro	RStudio Connect	shinyapps.io
URL-encoded bookmarking	✔	✔	✔	✔
Saving state to server	✔	Coming soon	In development	In development

How to bookmark

SHINY VERSION

`shiny v0.14` or higher

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, input$txt, input$txt)
  })
}

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

UI portion must be a function that takes one argument

There must be a call to enableBookmarking()

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, input$txt, input$txt)
  })
}

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

Only change for saving to server

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)
  - To exclude other inputs from being bookmarked, call `setBookmarkExclude()` in the server function, and pass in a vector containing the names of the inputs

```
# Server function
function(input, output, session) {
  setBookmarkExclude(c("x", "y"))
}
```

CUSTOM BOOKMARKING

- If the application state is entirely contained in the input widgets (i.e. data flow goes from inputs to (optional) reactives to outputs), Shiny will handle all bookmark/restore operations automatically.
- If the state of the inputs alone does not fully determine the state of the outputs (i.e. there's independent state in server-side variables), you must help Shiny by providing custom callback functions — more on this in *Advanced Bookmarking* section

UI CONCERNS

- It is possible to bookmark which tab in a tabset is active
  - This requires providing IDs for `tabsetPanel()`, `navbarPage()`, or `navlistPanel()`
- Generally, restoring apps with inputs in `renderUI` "just works" (let us know if not!)

bookmark.R

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()**.

Advanced bookmarking

“advanced” → apps that aren't purely reactive – where the state of the inputs at a given time doesn’t fully determine the state of the outputs

↓

use additional tools to save & restore desired state

adv_bookmark.R

DEMO

Displays the sum of all the previous slider values

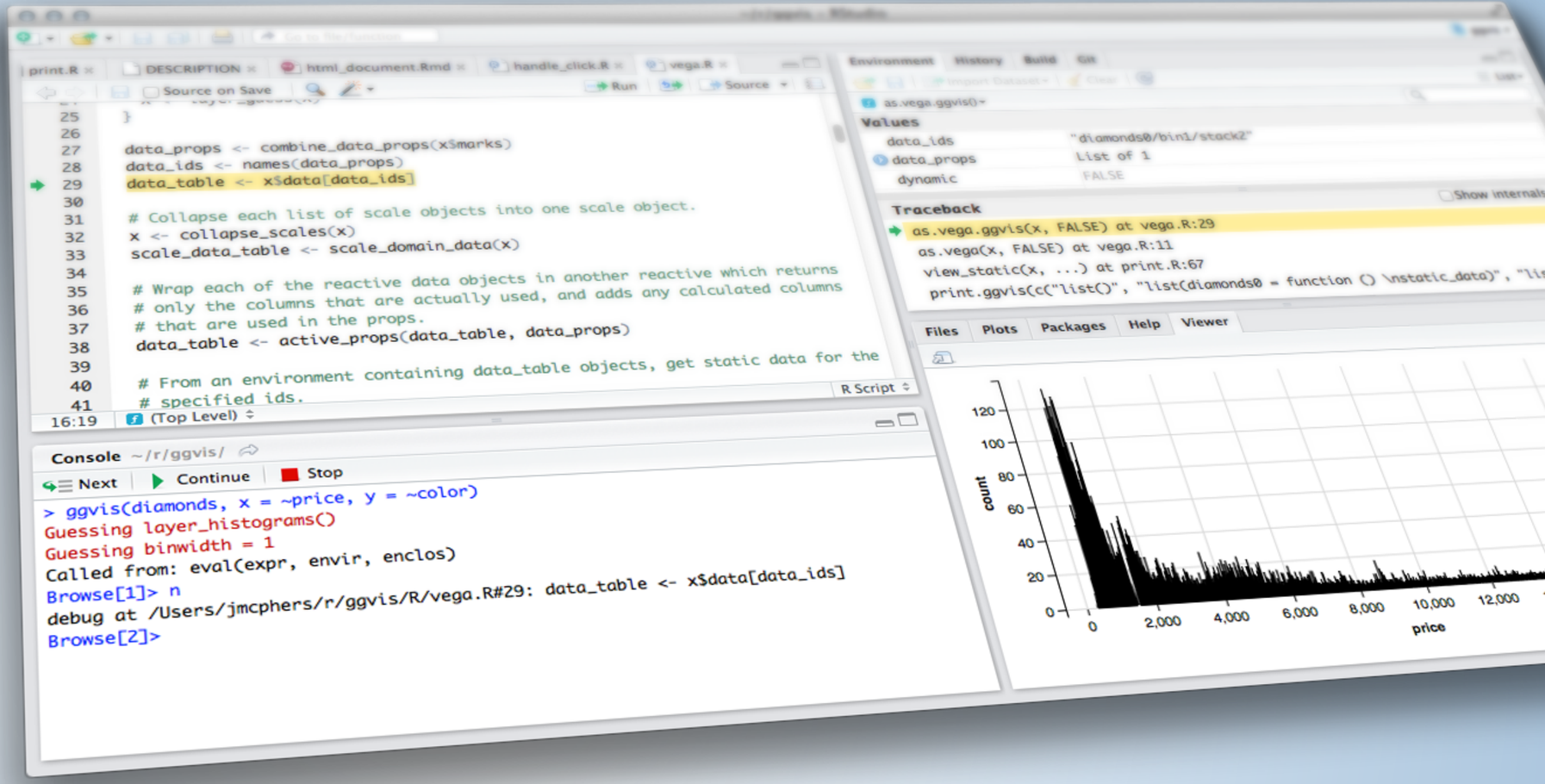
```
ui <- function(request) {
  fluidPage(
    sliderInput("slider", "Add a value:", 0, 100, 0),
    h4("Sum of all previous slider values: "),
    verbatimTextOutput("sum")
  )
}

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

  observe({
    # Use isolate() so that this observer doesn't invalidate itself
    vals$s <- isolate(vals$s) + input$slider
  })

  output$sum <- renderText({
    vals$s
  })
}

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

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

Bookmarking this application requires also storing previous input values as well

DEMO

```
ui <- function(request) {
  fluidPage(
    sliderInput("slider", "Add a value:", 0, 100, 0),
    h4("Sum of all previous slider values: "),
    verbatimTextOutput("sum")
    bookmarkButton()
  )
}

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

  observe({
   vals$s <- isolate(vals$s) + input$slider
  })

  output$sum <- renderText({
    vals$s
  })

  # Save extra values in state$values when we bookmark...
  onBookmark(function(state) {
    state$values$s <- vals$s
  })
  # Restore extra values from state$values when we restore
  onRestore(function(state) {
    vals$s <- state$values$s
  })
}

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



BOOKMARKABLE STATE

Shiny from RStudio™