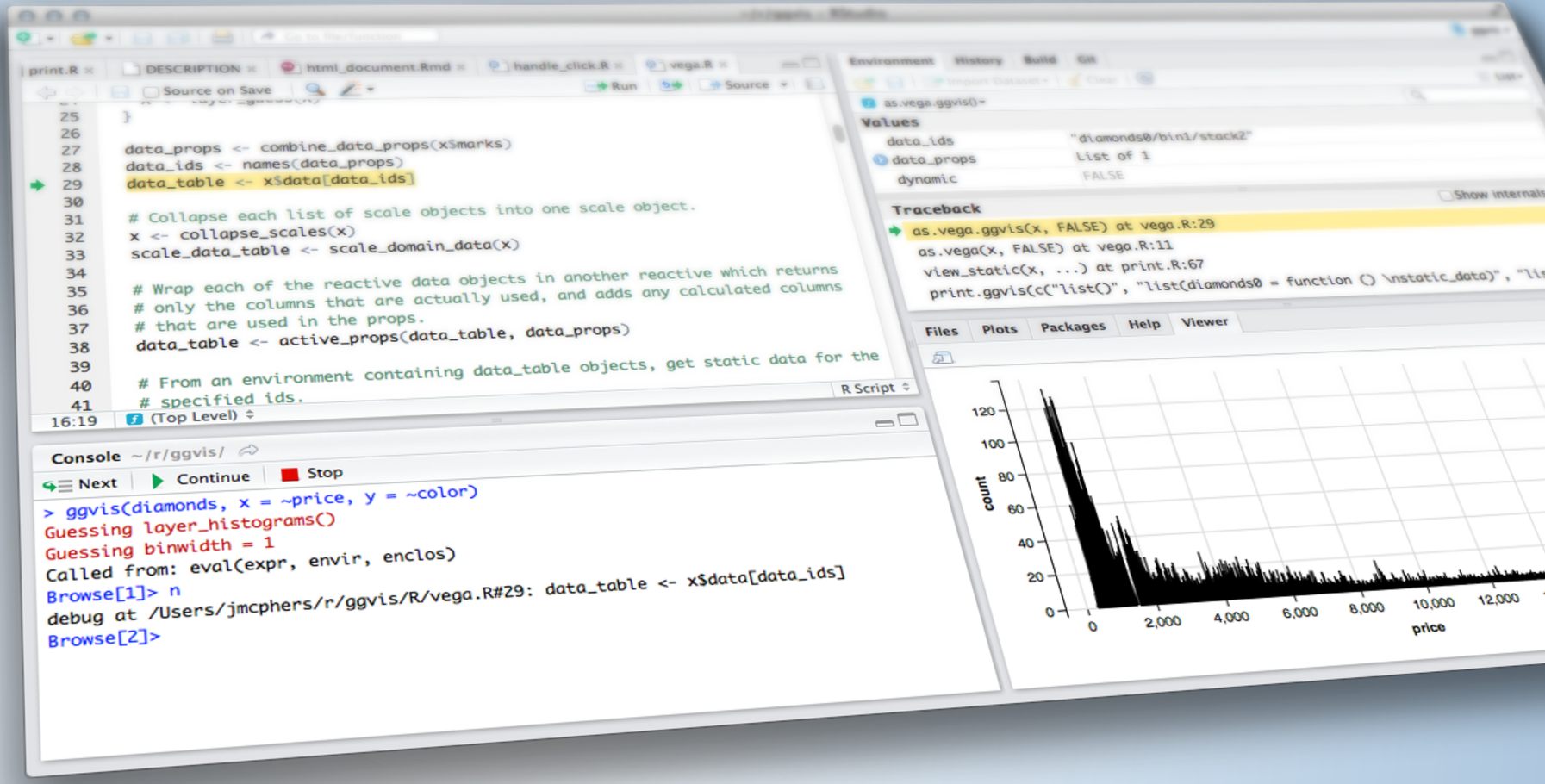


INTERACTIVE DATA VISUALIZATIONS

Shiny from RStudio™

OUTLINE

- Motivation
- Common actions
- Linking plots
  - Linked zooming
  - Linked brushing
  - Get creative

Motivation

MOTIVATION

- Use Shiny to create interactive data visualizations for data exploration as well as for presentation
- Exploration: Streamline first steps of your analysis where you might consider which data points to include in / exclude from your analysis, identify observations with interesting / outlying features, etc.
- Presentation: Create visualizations for communicating your findings in the same platform that you do your analysis

Common actions

COMMON ACTIONS

- Click
- Double-click
- Hover
- Brush

EXAMPLE CONTEXT

- The Southern Oscillation Index (SOI) is a standardized index based on the observed sea level pressure differences between Tahiti and Darwin, Australia.
- SOI is one measure of the large-scale fluctuations in air pressure occurring between the western and eastern tropical Pacific during El Niño and La Niña episodes.
  - Prolonged periods of **negative** SOI values coincide with abnormally **warm** ocean waters across the eastern tropical Pacific typical of **El Niño** episodes.
  - Prolonged periods of **positive** SOI values coincide with abnormally **cold** ocean waters across the eastern tropical Pacific typical of **La Niña** episodes.
- In general, smoothed time series of the SOI correspond very well with changes in ocean temperatures across the eastern tropical Pacific.
- Source: National Centers for Environmental Information

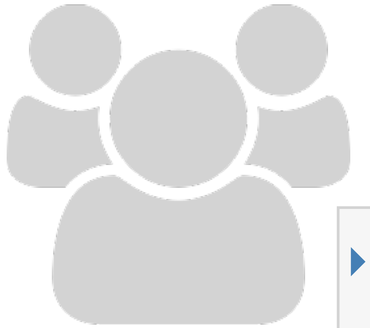
sol_01.R

DEMO

```
# UI

plotOutput("plot",
           click = clickOpts(id = "plot_click"),
           brush = brushOpts(id = “plot_brush")
)

# Server

output$click_info <- renderPrint({
  nearPoints(soi, input$plot_click) %>%
    mutate(month = month(month, label = TRUE)) %>%
    rename(SOI = value) %>%
    select(month, year, SOI)
})
```

Define user interaction types in the UI

Determine points user is interacting with in the server



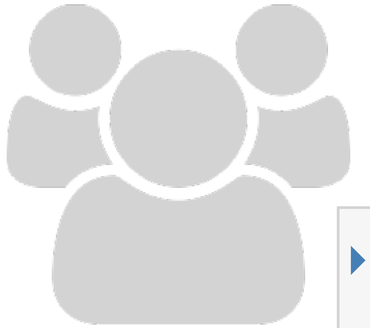
EXERCISE

- In `soi_01.R` the `verbatimTextOutput` reports `<0 rows>` before the user clicks / brushes for the first time. This behavior is undesirable, it makes the output messy and potentially confusing.
- Implement a fix for this such that the `verbatimTextOutput`, the values of the clicked / brushed points, are reported only after the user takes these actions.

3m 00s

Solution to the previous exercise

`soi_02.R`



EXERCISE

- There are two more common actions when interacting with data visualizations: double-click and hover
- Build on `soi_02.R` to implement these inputs, and add two more columns to the UI to report features on points double-clicked or hovered on
- Hint: See `?dblclickOpts` and `?hoverOpts`. Also note that the convenience function nearPoints can be used with click, double click, and hover.

5m 00s

Solution to the previous exercise

`soi_03.R`

Linking plots

Linked zooming

soi_04.R

DEMO

```
# UI
plotOutput("zoom", height = "350px"),
plotOutput("overall", height = "150px",
           brush =  brushOpts(id = "brush", direction = "x")

# Server
output$zoom <- renderPlot({
  if (!is.null(input$brush)) {
    p <- p +
      xlim(input$brush$xmin, input$brush$xmax) +
      labs(title = "Southern Oscillation Index (SOI)")
  }
  p
})

output$overall <- renderPlot(p)
```

Two plots, one interactive for zooming

Limit x and y for zoomed in plot

Linked brushing

soi_05.R

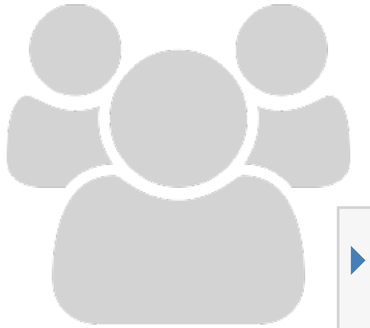
DEMO

```
# UI
plotOutput(“scatterplot", brush = brushOpts(id = "brush")
plotOutput("histogram")

# Server
output$scatterplot <- renderPlot({
  # standard plot
})

output$histogram <- renderPlot({
  brushed <- brushedPoints(soi, input$brush)
  # standard plot
})
```

Two plots, one interactive for brushing

Limit x and y for zoomed in plot

Get creative



EXERCISE

- Use the diamonds dataset in the ggplot2 package to create an app with at least two interactive data visualizations that have linked brushing and/or linked zooming between them.
- Keep it simple initially, if you finish early, continue to add more bells and whistles to your app.

10m 00s