BUILDING WEB APPLICATIONS IN R WITH SHINY

Stop-trigger-delay

Isolating reactions

Goal: Update plot (and title) when inputs other than `input$plot_title` changes.

```
output$scatterplot <- renderPlot({
  ggplot(data = movies_subset(), aes_string(x = input$x, y = input$y)) +
    geom_point() +
    labs(title = isolate({ input$plot_title }) )
})
```

Plot title will update when any of the other **`input`**s in this chunk change

Plot title will **not** update when `input$plot_title` changes

Triggering reactions

expression to call whenever `eventExpr` is invalidated

```
observeEvent(eventExpr, handlerExpr, …)
```

simple reactive value - `input$click`,
call to reactive expression - `df()`,
or complex expression inside `{}`

Triggering reactions

Goal: Write a CSV of the sampled data when action button is pressed.

```
# ui
actionButton(inputId = "write_csv", label = "Write CSV")
```

```
# server
observeEvent(input$write_csv, {
             filename <- paste0("movies_",
                                str_replace_all(Sys.time(), ":|\ ", “_"),
                                ".csv")
             write_csv(movies_sample(), path = filename)
             }
)
```

Delaying reactions

expression to call whenever `eventExpr` is invalidated

```
eventReactive(eventExpr, handlerExpr, …)
```

simple reactive value - `input$click`,
call to reactive expression - `df()`,
or complex expression inside `{}`

Delaying reactions

Goal: Change how the random sample is generated such that it is updated when the user clicks on an action button that says “Get new sample”.

```
# ui
actionButton(inputId = "get_new_sample", label = "Get new sample")
```

```
# server
movies_sample <- eventReactive(input$get_new_sample,{
                                  req(input$n_samp)
                                  sample_n(movies_subset(), input$n_samp)
                                  },
                                  ignoreNULL = FALSE
)
```

Initially perform the action/calculation and just let the user re-initiate it (like a "Recalculate" button)

observeEvent vs. eventReactive

- observeEvent() is to to perform an action in response to an event
- eventReactive() is used to create a calculated value that only updates in response to an event

observeEvent/eventReactive vs. observe/reactive

- observe() and reactive() functions automatically trigger on whatever they access
- observeEvent() and eventReactive() functions need to be explicitly told what triggers them

isolate vs. event handling functions

- isolate() is used to stop a reaction
- observeEvent() is used to perform an action in response to an event
- eventReactive() is used to create a *calculated value* that only updates in response to an event

BUILDING WEB APPLICATIONS IN R WITH SHINY

Let's practice!