Shiny
Visualizacion de datos con R
Slides 4

Diplomado en Data Science, MatPUC

Joshua Kunst Fuentes jbkunst@gmail.com https://jkunst.com/blog/

Shiny: Visualizacion de datos con R

Version 2025

Clases

- Martes 18 de noviembre
- Jueves 20 de Noviembre

Programa

- Slides 1
  - Aplicación (web), ejemplos.
  - Introducción a shiny: Interfaz usuario y servidor
- Slides 2
  - Layouts
  - Integración HTMLWidgets
- Slides 3
  - Temas, templates y diseño
  - Compartir una app
  - Shiny en Python
- Slides 4
  - Expresiones reactivas
  - Orden de ejecución
  - Extensiones shiny

Ejercicio: Revisar application

Revise la aplicación del ejercicio "inputs + laytous + htmlwidgets" que utilizaba la función `descargar_data`.

- Que sucede en cada bloque `render`?
- Que pasa si tuviera más outputs (y por tanto renders) que utilicen estos datos?
- Modifique la función `descargar_data` agregnado al inicio del cuerpo de la funcion `Sys.sleep(5)` y pruebe la aplicación con esta modificación.

Expresiones reactivas

Expresiones reactivas (*reactive expressions*)

La idea de expresiones reactiva es que podemos limitar que es lo que se (re)ejecuta al cambiar un input.

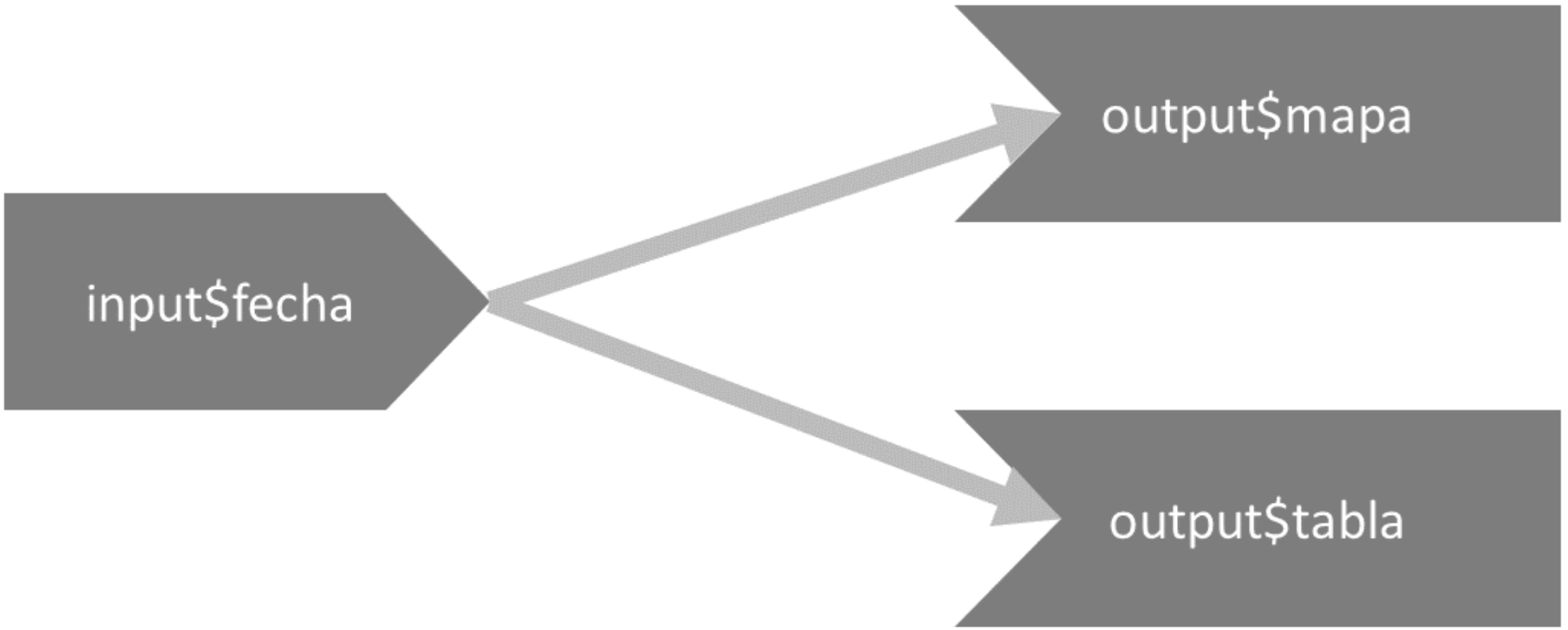
Una expresión reactiva es código R que usa un input y retorna un valor, la expresion se actualizará cuando el valor del (de los) inputs de los cuales dependen cambien.

Estos elementos se definen en el `server`. Se crea una expresion con la función `reactive` la que toma una expresión/código R entre `{}`, de la misma forma que las funciones `render` (`renderPlot`, `renderTable`):

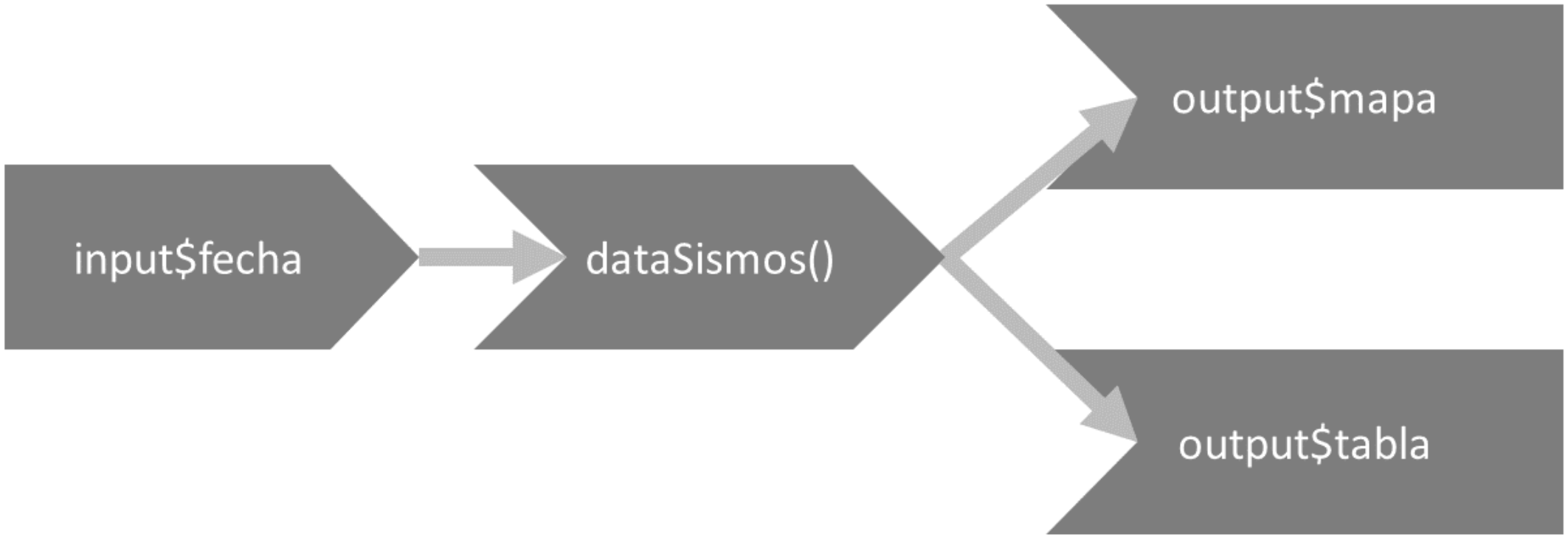
```
server <- function(input, output) {
  ...
  elementoReactivo <- reactive({
    codigo R...
  })
  ...

}
```

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

  dataSismos <- reactive({
    fecha <- ymd(input$fecha)
    ...
    datos <- read_html(url) |>
      html_table() |> ...
    datos

  })

  output$mapa <- renderLeaflet({

    datos <- dataSismos()

    leaflet(datos) |> ...
  })

  output$tabla <- renderDT({
    datatable(dataSismos())
  })

})
```

Expresiones reactivas: Ejemplo



Expresiones reactivas: Ejemplo *mejorado*



Ejercicio: Creando una expresión reactiva

Para el ejercicio "inputs + laytous + htmlwidgets":

- Cree la expresión reactiva necesaria para reducir la cantidad de descargas.

- Se puede mejorar? (`bindCache`)

Orden de ejecución

Una vez

```
# A place to put code

ui <- fluidpage(

)

server <- function(input, output) {

  # Another place to put code

  output$map <- renderPlot({

    # A third place to put code

  })
}

shinyApp(ui, server)
```

Run once when app is launched

Una vez por usuario

```
# A place to put code

ui <- fluidpage(

)

server <- function(input, output) {

  # Another place to put code

  output$map <- renderPlot({

    # A third place to put code

  })
}

shinyApp(ui, server)
```

Run once each time a user visits the app

Muchas veces

```
# A place to put code

ui <- fluidpage(

)

server <- function(input, output) {

  # Another place to put code

  output$map <- renderPlot({

    # A third place to put code

  })
}

shinyApp(ui, server)
```

Run once each time a user changes a widget that output$map depends on

Extensiones para Shiny

Extensiones para Shiny

Podemos decir que ya existe un *shinyverso* dada la cantidad de paquetes que extienden shiny, agregando tanto diseños, nuevas funcionalidades, etc:

https://github.com/nanxstats/awesome-shiny-extensions

Menciones honrosas (*en mi opinión*):

- https://waiter.john-coene.com/

- https://cicerone.john-coene.com/

Buen ejercicio es incorporar las funcionalidades de {waiter} y {ciceroene} en la aplicación.

Cosas que no vimos 😔... Pero que existen! 😀

... Que ni tan necesarias, pero que en futuro podrían ayudar!

- `bindCache`: Se puede asociar un resultado de un `render` a ciertos inputs, para guardar automáticament el resultado sin tener que volver a ejecutar el código dentro del `render`. https://shiny.rstudio.com/articles/caching.html

- Bookmarking: La posibilidad de registrar la aplicación con ciertos inputs seleccionados. Como cuando comparter una aplicación de un retail con ciertos filtros. https://mastering-shiny.org/action-bookmark.html

- HTMLWidgets Proxys: Características de algunos HTMLWidgets para no *recontruir* el gráfico sino que actualizarlo. https://jbkunst.shinyapps.io/bias-variance/