{ SHINY II}

[Clase 3]

Clase 3.

Contenidos principales:

- Repaso de fundamentos de shiny
- Basic widgets: radio buttons, select input, check box y slider input.
- Graficas reactivas con Ggplot y eventos con click.
- Programacion reactiva: source, endpoint y conductor

Luego de esta clase serás capaz de:

- Generar una aplicación shiny con gráficas interactivas por medio de utilización widgets y gráficas en ggplot2.

DATASET GAPMINDER

En esta clase vamos a utilizar el dataset de GAPMINDER, el mismo corresponde:
“Un extracto de los datos disponibles en Gapminder.org. Para cada uno de los 142 países, el paquete proporciona valores para la esperanza de vida, el PIB per cápita y la población, cada cinco años, desde 1952 hasta 2007“

DATASET GAPMINDER

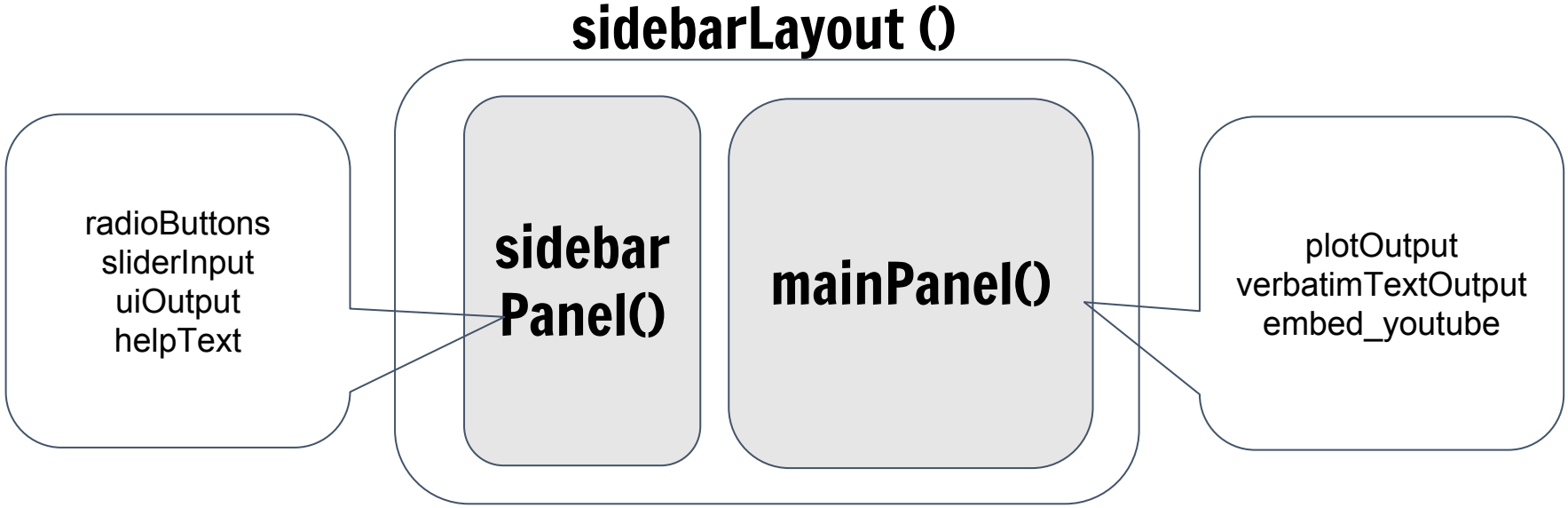
Para instalar en R el paquete:

install.packages('gapminder')

{ UI }

User interface

UI: FLUID PAGE

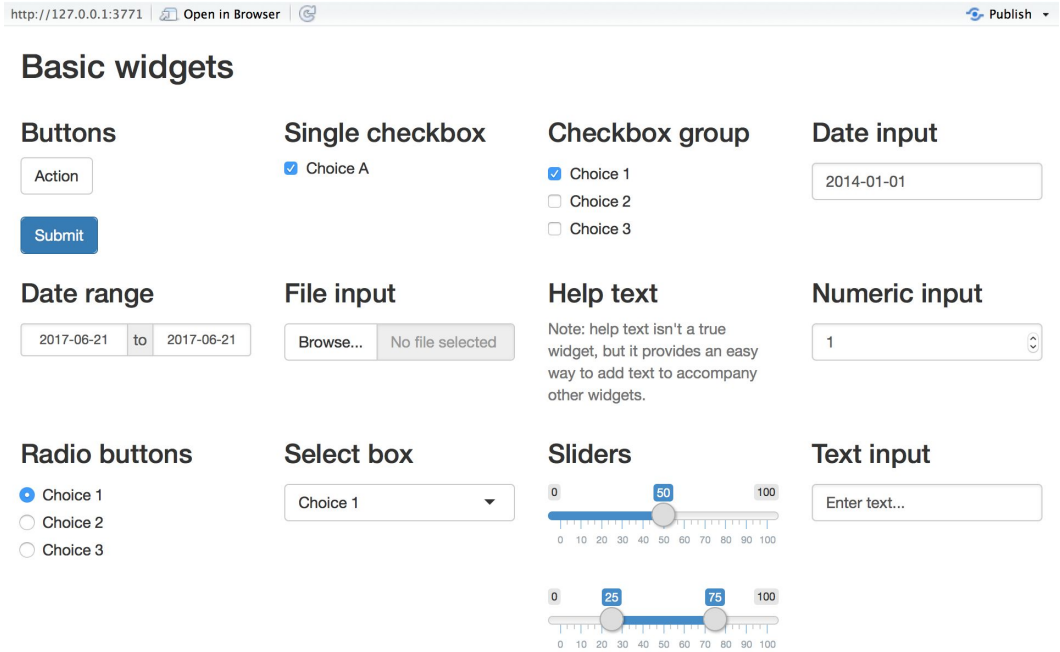


UI: FLUID PAGE

```
ui <- fluidPage (
  titlePanel('ANÁLISIS DE POBLACIÓN'),
  sidebarLayout (
     sidebarPanel (
                  #Widgets ),
      mainPanel (
          #Graficas, Texto, video, etc )
     ) )
```

{ input }

BASIC WIDGETS



UI: RADIO BUTTONS

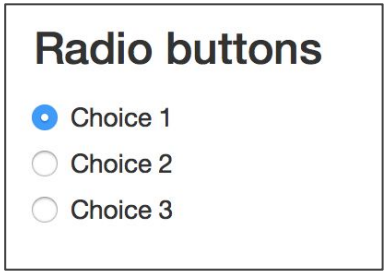
```
radioButtons(
    inputId='continente',
    label='selección de continente',
    choices = NULL,
    selected = NULL)
```

Id de la variable en el servidor

Leyenda que aparece en la UI

Valores posibles para seleccionar

Valor inicial seleccionado



Output : “Choise 1”

UI: SELECT INPUT

Puedo agregar una etiqueta de encabezado de html "h3"

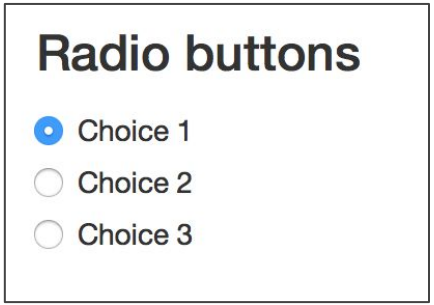
```
selectInput(
        inputId='continente',
        label= h3('selección de continente'),
        choices = NULL,
        selected = NULL)
```

Output : **"Choice 1"**

UI: RADIO BUTTONS - SELECT INPUT

choices=

- **c(‘var1’,’var2’,’var3)**
- **unique(df$var)**
- **colnames(df)**
- **rownames(df)**



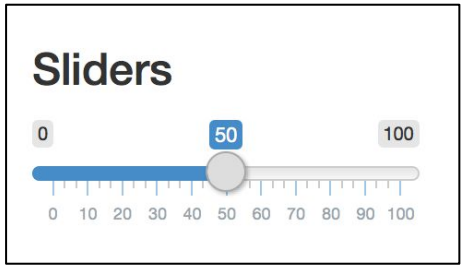
UI: SLIDERS

```
sliderInput(
        inputId=’valor’,
        label= h3('Selección del valor'),
        min = 0,
        max = 100,
        value = 30)
)
```

Valor maximo

Valor minimo

Valor inicial



Output : 30

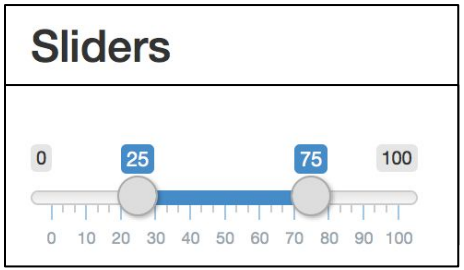
UI: SLIDERS

```
sliderInput(
        inputId=’valor’,
        label= h3('Selección del valor'),
        min = 0,
        max = 100,
        value = c(25,75)
)
```

Valor maximo

Valor minimo

Valor inicial



Output : “25” “75”

UI: SLIDERS

```
helpText(
        'Observaciones:',
        “Un extracto de los datos disponibles en Gapminder.org.
        Para cada uno de los 142 países, el paquete proporciona valores para la esperanza de vida, el PIB per cápita y la población, cada cinco años, desde 1952 hasta 2007."

)
```

UI: SLIDERS

```
checkboxInput(

    inputId= 'input_masinfo',
     label=strong('Mas info'),
     value = FALSE
)
```

Puedo utilizar strong para colocar énfasis

Output : **TRUE**

SHINY THEMES

```
install.packages('shinythemes')
```

```
themeSelector()
```

Para ver todos los temas..
https://bootswatch.com/

```
theme = shinytheme("____")
```

{ output }

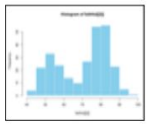
PLOT OUTPUT

UI function

plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)

```
plotOutput(
    inputid='g1',
    click = "plot_click")
```

Server function



renderPlot(expr, width, height, res, …, env, quoted, func)

VERBATIM TEXT OUTPUT

UI function

verbatimTextOutput(outputId)

Server function

```
verbatimTextOutput(
    inputid='info')
```

renderPrint(expr, env, quoted, func, width)

{ Practica}

Realizar la siguiente interfaz tomando en cuenta los input y output que aprendimos durante la clase y seleccionar un tema de shinythemes.

radiobutons ("input_continente"...)

checkboxInput (input_masinfo…)

sliderInput (input_fecha…)

selectInput (input_pais…)

helpText

ANALISIS DE POBLACION

seleccion de continente

- (•) Asia
- ( ) Europe
- ( ) Africa
- ( ) Americas
- ( ) Oceania

[x] **Mas info**

seleccion del periodo

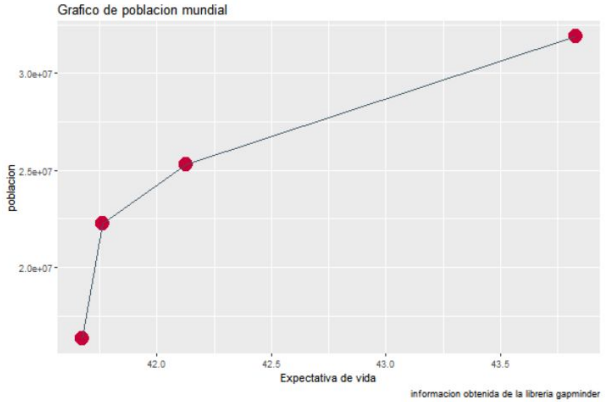
1,952 — 1,990 — 2,007

1,952 1,963 1,974 1,985 1,996 2,007

Seleccionar el pais

Afghanistan

Observaciones: Un extracto de los datos disponibles en Gapminder.org. Para cada uno de los 142 países, el paquete proporciona valores para la esperanza de vida, el PIB per cápita y la población, cada cinco años, desde 1952 hasta 2007.




```
Coordenadas:
x=
y=
```

GdpPercap primedio es: 746.5

```
    lifeExp          pop
 Min.   :41.67   Min.   :16317921
 1st Qu.:41.74   1st Qu.:20750042
 Median :41.95   Median :23747910
 Mean   :42.35   Mean   :23925916
 3rd Qu.:42.55   3rd Qu.:26923784
 Max.   :43.83   Max.   :31889923
```

PlotOutput (output_g1...)

verbatimTextOutput (output_coor...)

TextOutput (output_masinfo...)

verbatimTextOutput (summary...)

{ server }

UI Output - renderUI

UI function

uiOutput(outputId, inline, container, …)

uiOutput("combo_pais")

Server function

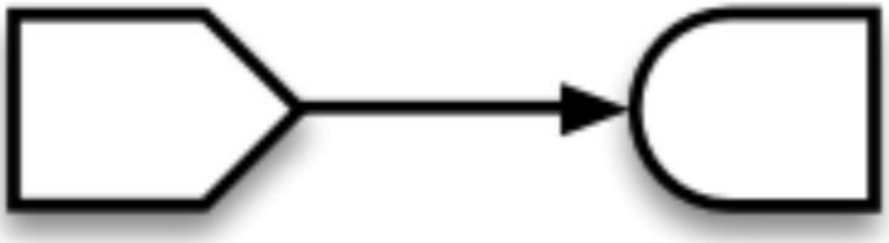
renderUI(expr, env, quoted, func)

```
output$combo_pais <- renderUI ({
selectInput (
   inputId="output_pais",
   label=h3("Seleccionar el pais"),
   choices = unique(
         df [ df $ continent == input$input_continente
            ,'country' ]) ,
   selected = 1
         )
 })
```

REACTIVE

Una expresión reactiva se define como aquella que transforma las entradas reactivas en salidas reactivas. Una app shiny permite realizar una programación reactiva en la cual el usuario puede realizar cambios en la salidas sin necesidad de modificar el código.

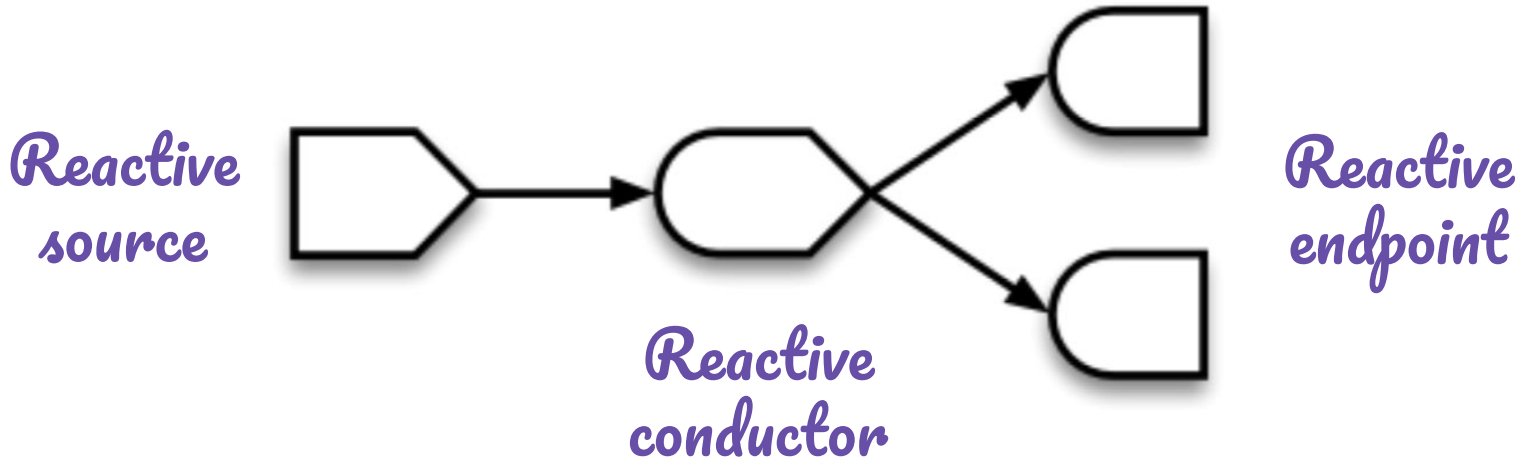
REACTIVE SOURCES AND ENDPOINTS



Reactive source

Reactive endpoint

REACTIVE CONDUCTORS



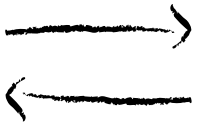
REACTIVE CONDUCTOR

```
df.filt <- reactive ({
 df.filt = df [
    df$country == input$input_pais          &
    df$year    >=  input$input_fecha[1]     &
    df$year    <=  input$input_fecha[2]      ,    ]
  df.filt
})
```

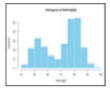
PLOT OUTPUT - RENDER PLOT

UI function

plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)



Server function



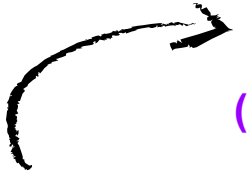
renderPlot(expr, width, height, res, …, env, quoted, func)

```
plotOutput(
      inputid='output_g1',
      click = "plot_click")
```

```
output$output_g1=_______({
  ggplot(____,
       aes( x = lifeExp, y = pop )) +
      geom_point (size=6, color = ”red” )+
      geom_line(color='_____')+
      labs(x='__ ', y='__ ', title='_________ ',
          caption='_____________ ’ )
})
```

VERBATIM TEXT OUTPUT

Server function



UI function
verbatimTextOutput
(inputid="output_coor")

```
output$output_coor <- renderText({
  paste0("Coordenadas:",
         "\nx=", input$plot_click$x,
         "\ny=", input$plot_click$y)
})
```

VERBATIM TEXT OUTPUT

Server function

```
output$output_coor <- renderText({
  paste0("Coordenadas:",
         "\nx=", input$plot_click$x,
         "\ny=", input$plot_click$y)
})
```

UI function

```
plotOutput
(inputid='output_g1',
click = "plot_click")
```

{ Practica}

Realizar el desarrollo del server con:

1. renderUI: output$combo_pais, generando un control selectinput que dependa de la elección del continente realizado.

2. reactive: crear df.filt que filtra el dataset en base a:
   - ➔ Country == input$output_pais
   - ➔ year>=input$input_fecha[1]
   - ➔ year<=input$input_fecha[2]

3. renderPlot: output$output_g1, crear un gráfico ggplot con:
   - Dataset: df.filt()
   - X = lifeExp
   - Y = pop

Realizar el desarrollo del server con:

3. renderPlot: output$output_g1, crear un gráfico ggplot con:
    - geom_point(size=6,color =’xxx’) +
      geom_line(color='#3F5866')
    - labs (x='',y='',title='',caption='')

4. renderText: output$output_coor, crear una acción para que cuando se realice un click sobre un punto muestre las coordenadas del punto

```
paste0("Coordenadas:", "\nx=", input$plot_click$x,
       "\ny=", input$plot_click$y)
```

Realizar el desarrollo del server con:

5. renderPrint: output$summary, realizar el summary del dataset df.filt() y que solo lo realice sobre las columnas 'lifeExp' y 'pop'

```
{ Recueda: summary( dfnombre[, c(‘col1’,‘col2’)] )  }
```

6. renderText: output_masinfo, en el caso que el checkbox retorne TRUE imprimir en el verbatimtext el promedio de la variable GdpPercap con un texto concatenado:

GdpPercap primedio es: ‘xxxx.xx’

```
  output$output_masinfo <- renderText({

    if( input$input_masinfo == TRUE) {

        prom = mean ( df.filt() $ gdpPercap )
        paste0 ('GdpPercap promedio es: ',round(prom,2) )

    }

  })
```

{ video }

```
install.packages('vembedr')
```

En la UI puedes agregar:

```
embed_youtube( "PUwmA3Q0_OE" ,width=420, height=315)
```

https://www.youtube.com/watch?v= **PUwmA3Q0_OE** &t=13s