

A collection of approximately 15 squares in light blue, medium blue, and grey, scattered across the top half of the slide.

MUBD

Màster Universitari en Enginyeria de Dades Massives (Big Data)

Estadística

Índice

1. Scripts necesarios: ui.R y server.R

2. ui.R

a. Estructura básica

b. Widgets (tipos de inputs)

c. Tipos de output

3. Server.R

a. Estructura básica

b. Tipos de output

Shiny

Ejemplo

<http://shiny.rstudio.com/gallery/kmeans-example.html>

Shiny by RStudio

BACK TO GALLERY

GET CODE

SHARE

Search

Iris k-means clustering

X Variable

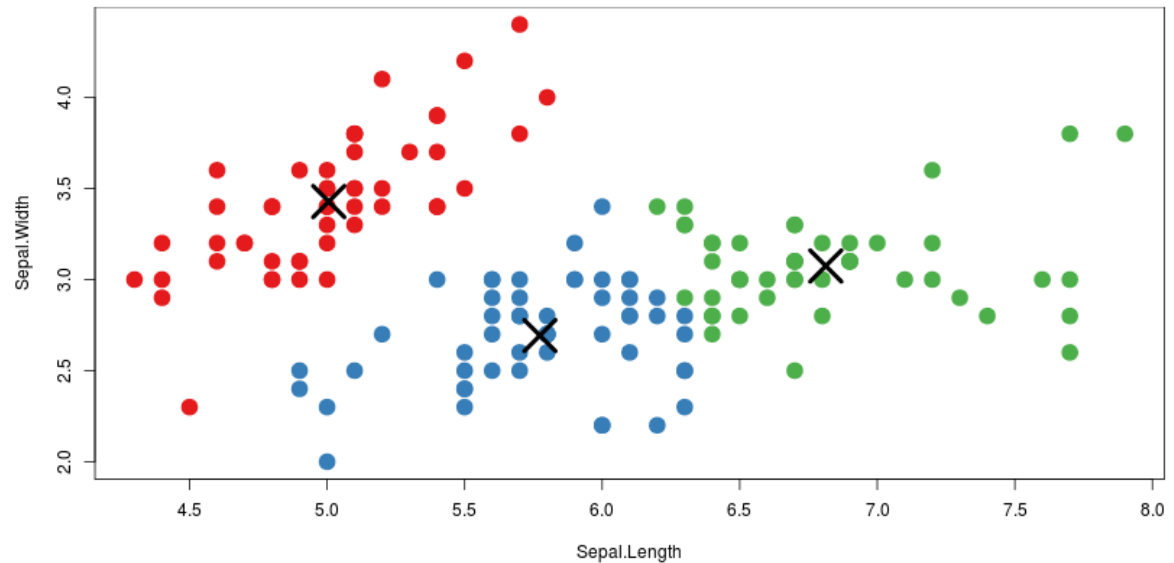
Sepal.Length

Y Variable

Sepal.Width

Cluster count

3



Shiny

Introducción

- Paquete de R para crear aplicaciones web
- Para apps simples, sin necesidad de conocimiento de HTML ni javascript
- Enlaces:
 - Tutorial: <http://shiny.rstudio.com/tutorial/>
 - Galería: <http://shiny.rstudio.com/gallery/>

Shiny

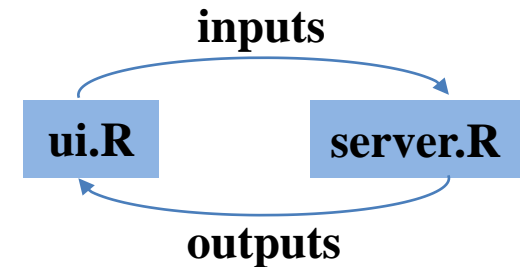
Scripts necesarios

■ ui.R

- Especificación de la “**user interface**”
- Debe contener *inputs* donde recoger la información proporcionada por el usuario. Estos *inputs* se enviarán al server
- Debe contener *outputs* procedentes del server

■ server.R

- Especificación de las acciones que realiza la app
- Recibe *inputs* del *ui* y envía *outputs*



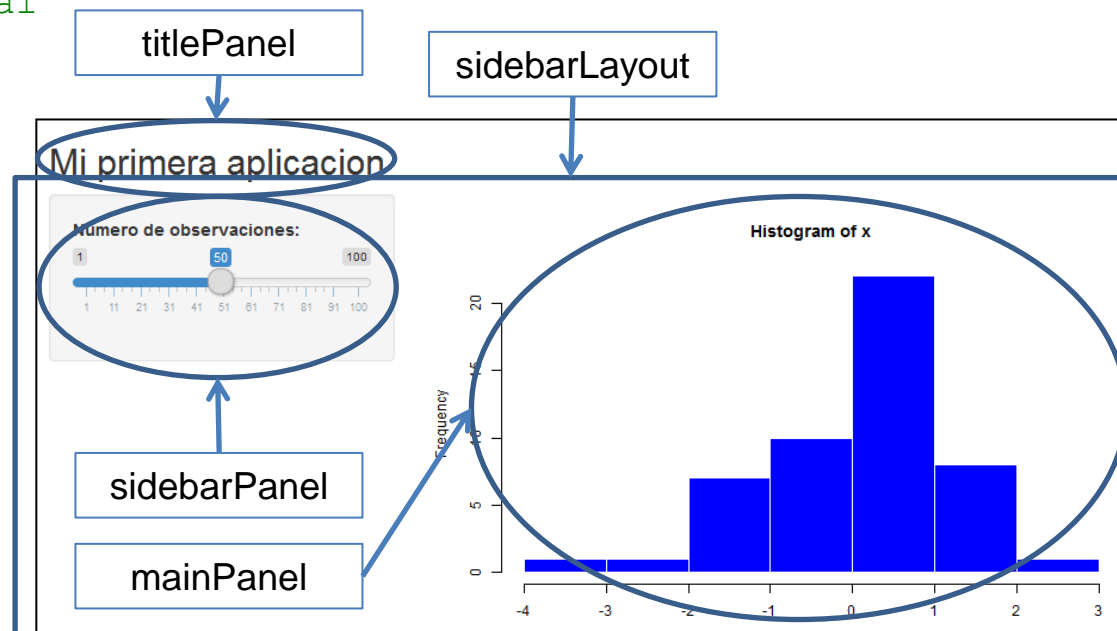
User Interface (ui)

Estructura básica

```
library(shiny)

shinyUI(fluidPage(  # fluidPage: incluye filas o estructura determinada (sideBar)
  titlePanel("..."), # titulo de la app
  sidebarLayout(     # estructura de la App
    sidebarPanel(
      ...    # inputs en el lateral
    ),
    mainPanel(
      ... # Panel de resultados
    )
  )
))
```

Atención a la sintaxis: dejarse comas o paréntesis son errores muy frecuentes



User Interface (ui)

Widgets

- Los widgets son los elementos de la ui que nos sirven para introducir los inputs. Las funciones para los más relevantes son:
 - Selector: `selectInput`
 - Número: `numericInput`
 - Texto: `textInput`
 - marcador con 2 valores posibles: `checkboxInput`
 - Todos: <http://shiny.rstudio.com/gallery/widget-gallery.html>
- Cada widget tiene algunos parámetros necesarios para especificar sus características. El primero de ellos siempre es el nombre que se le dará a ese input. Ejemplo:

```
numericInput('num', label='Número de grupos:', min = 2, max = 10)
```

User Interface (ui)

Tipos de output

- Shiny tiene funciones que transforman los outputs del server en contenido adaptable a nuestra user-interface:
 - *Figuras*: `plotOutput`
 - *Tablas*: `tableOutput`
 - *Texto (o HTML)*: `textOutput`
 - Todos: <http://shiny.rstudio.com/tutorial/lesson4/>
- Necesitaremos una función para cada output
- Ejemplo: el objeto *grafico* proviene del server

```
mainPanel(  
  plotOutput("grafico")  
)
```


server

Estructura básica

```
library(shiny)

shinyServer(function(input, output) {

    ...                # introducir las funciones para generar los outputs

})
```

Ejemplo

```
library(shiny)

shinyServer(function(input, output) {

    output$plot1 <- renderPlot({ # "plot1" será el output que recibirá el ui
        hist(rnorm(input$value)) # input$value contiene el valor recibido del ui
    })

})
```

server

Outputs

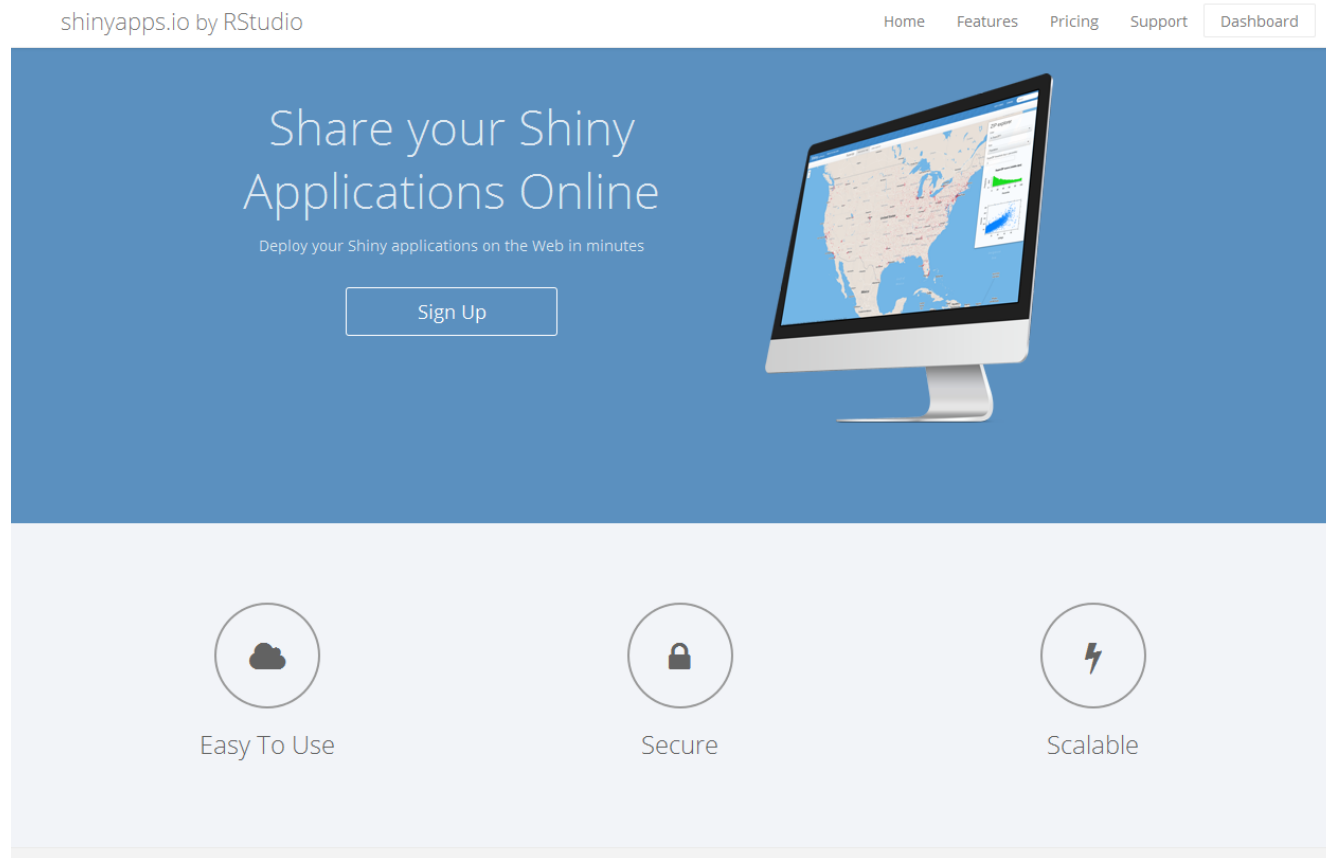
- El server devuelve objetos renderizados con funciones en la que hemos de especificar el tipo de salida
 - Figuras: `renderPlot`
 - Tablas: `renderTable`
 - Texto: `renderText`
- Ejemplo. Un output que sea un gráfico de un boxplot

```
output$bxp <- renderPlot({  
  x <- 1:20  
  boxplot(ct, main=input$tit)  
})
```

Shiny

¿Dónde hacer visibles nuestras apps?

- <https://www.shinyapps.io/>





MUBD

Màster Universitari en Enginyeria de Dades Massives (Big Data)

Estadística

