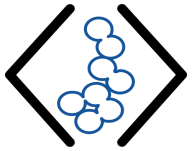


Shiny

Sesión II: Extensiones a Shiny

Hèctor Perpiñán Fabuel - Unitat de Bioestadística, IRBLLEIDA

07 de Junio, 2017

Contenido

1. Customizar apariencia
   - HTML & CSS
2. Extensiones a Shiny (mediante paquetes)
3. Compartir las apps por internet
4. MapEs (una app desarrollada por FISABIO - DG Salud Pública)

1. Customizar apariencia

HTML & CSS

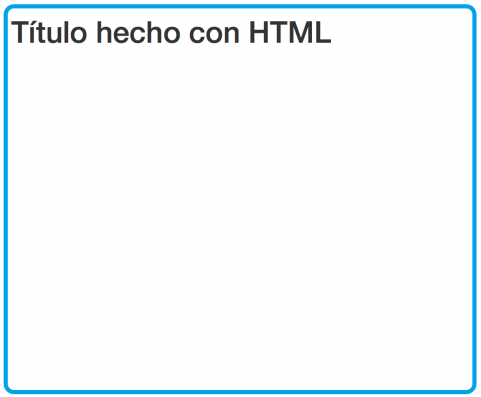
Comandos HTML

Shiny acepta código HTML:

```
ui <- fluidPage(
  HTML("<h1>Título hecho con HTML</h1>")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

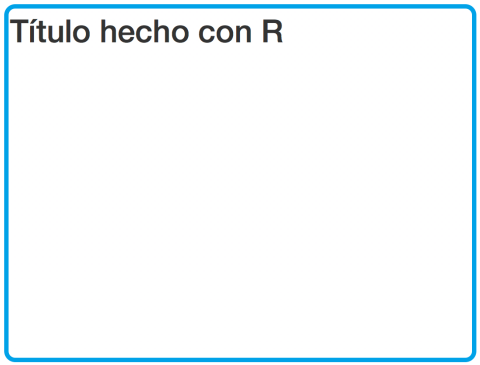


R permite añadir contenido a una página con las funciones **tags**

```
ui <- fluidPage(
  tags$h1("Título hecho con R")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```



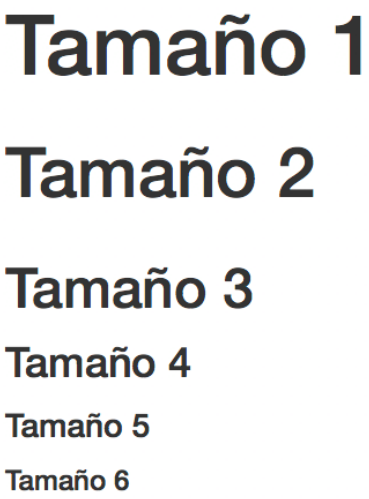
tags: **h1() - h6()**

Cabeceras

```
ui <- fluidPage(
  h1("Tamaño 1"),
  h2("Tamaño 2"),
  h3("Tamaño 3"),
  h4("Tamaño 4"),
  h5("Tamaño 5"),
  h6("Tamaño 6")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```



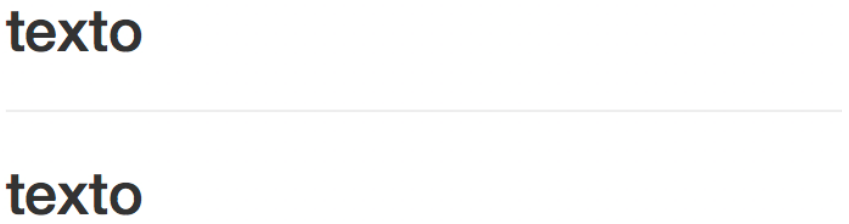
tags: **hr()**

Línea horizontal

```
ui <- fluidPage(
  h3("texto"),
  hr(),
  h3("texto")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```



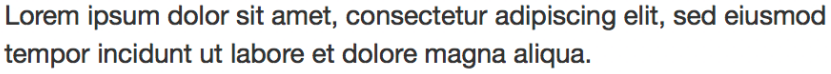
tags: "**Texto**"

Texto normal. El texto plano sin modificadores no necesita **tags**.

```
ui <- fluidPage(
  "Lorem ipsum dolor sit amet, consectetur adipisci
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```



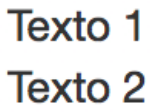
tags: **br()**

Salto de línea

```
ui <- fluidPage(
  "Texto 1",
  br(),
  "Texto 2"
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```



tags: **p()**

Párrafo

```
ui <- fluidPage(
  p("Párrafo 1"),
  p("Párrafo 2")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

tags: **em()**

Itálica

```
ui <- fluidPage(
  em("itálica")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

itálica

tags: **strong()**

Negrita

```
ui <- fluidPage(
  strong("negrita")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

negrita

tags: **code()**

Texto monoespaciado

```
ui <- fluidPage(
  code("código")
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

`código`

Los **tags** se pueden anidar unos dentro de otros.

```
ui <- fluidPage(
  p("Lorem ipsum dolor sit amet, ", strong("consect
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

Lorem ipsum dolor sit amet, **consectetur** adipiscing elit, *sed eiusmod tempor incidunt ut labore et dolore magna aliqua.*

Insertar un CSS

Las hojas de estilo en cascada (CSS) son un marco para personalizar la apariencia de elementos en una página web.

```
ui <- fluidPage(
  theme = "bootstrap.css",
  sidebarLayout(
    sidebarPanel(),
    mainPanel()
  )
)

# Definición del server
server <- function(input, output) {}

# App completa con los componentes ui y server
shinyApp(ui, server)
```

```
ui <- fluidPage(
  includeCSS("bootstrap.css"),
  sidebarLayout(
    sidebarPanel(),
    mainPanel()
  )
)

# Definición del server
server <- function(input, output) {}
```

2. Extensiones a Shiny (mediante paquetes)

2. Extensiones a Shiny (mediante paquetes)

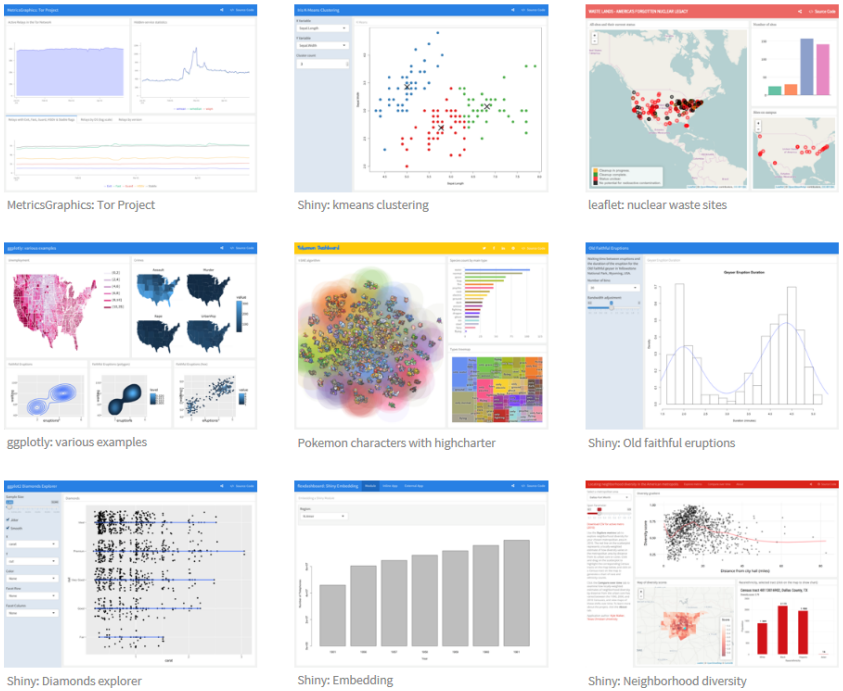
- Extensiones de formato
  - **flexdashboard**: Tableros interactivos fáciles para R (RMarkdown)
  - **shinythemes**: Temas CSS listos para usar con Shiny
  - **shinydashboard**: Tableros para Shiny
  - **shinyjqui**: Interacciones y efectos de animación para Shiny
- Extensiones para cálculos/gráficos interactivos
  - **htmlwidgets**: Un marco para embeber visualizaciones de JavaScript en R

flexdashboard

(http://rmarkdown.rstudio.com/flexdashboard/index.htm

- Formalmente es un **RMarkdown** (documento interactivo) con elementos de **Shiny**
- Muy buena combinación con **Github**
- Redes sociales (Twitter, Facebook, Google+, LinkedIn and Pinterest)
- Posibilidad de incrustar código

(https://beta.rstudioconnect.com/jjallaire/l showcase-storyboard/htmlwidgets-showcase-storyboard.html)

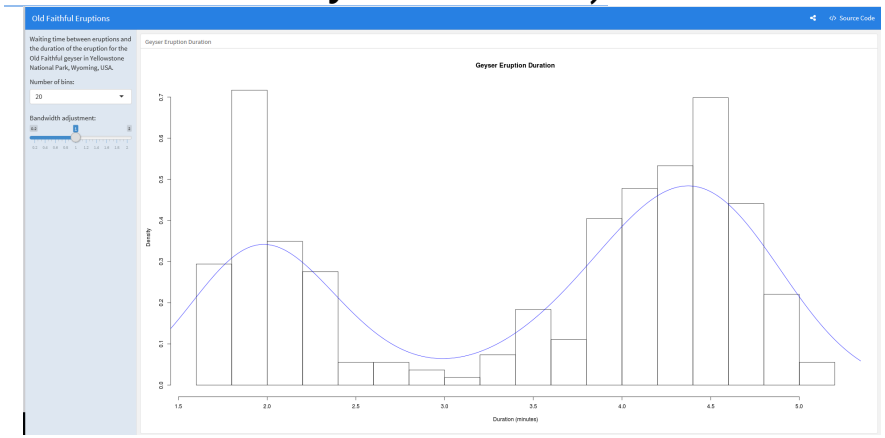


(https://beta.rstudioconnect.com/jjallaire/htm showcase-storyboard/htmlwidgets-showcase-storyboard.html)

flexdashboard

(http://rmarkdown.rstudio.com/flexdashboard/index.htm

(https://beta.rstudioconnect.com/jjallaire/l
showcase-storyboard/htmlwidgets-
showcase-storyboard.html)



(https://beta.rstudioconnect.com/jjallaire/l
showcase-storyboard/htmlwidgets-
showcase-storyboard.html)
(https://beta.rstudioconnect.com/jjallaire/s
eruptions/)

```
---
title: "Old Faithful Eruptions"
output:
  flexdashboard::flex_dashboard:
    orientation: rows
    social: menu
    source_code: embed
runtime: shiny
---

```{r global, include=FALSE}
# load data in 'global' chunk so it can be shared by all users of the dashboard
library(datasets)
data(faithful)
```

Column {.sidebar}

Waiting time between eruptions and the duration of the eruption for the
Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

```{r}
selectInput("n_breaks", label = "Number of bins:",
            choices = c(10, 20, 35, 50), selected = 20)

sliderInput("bw_adjust", label = "Bandwidth adjustment:",
            min = 0.2, max = 2, value = 1, step = 0.2)
```

Column

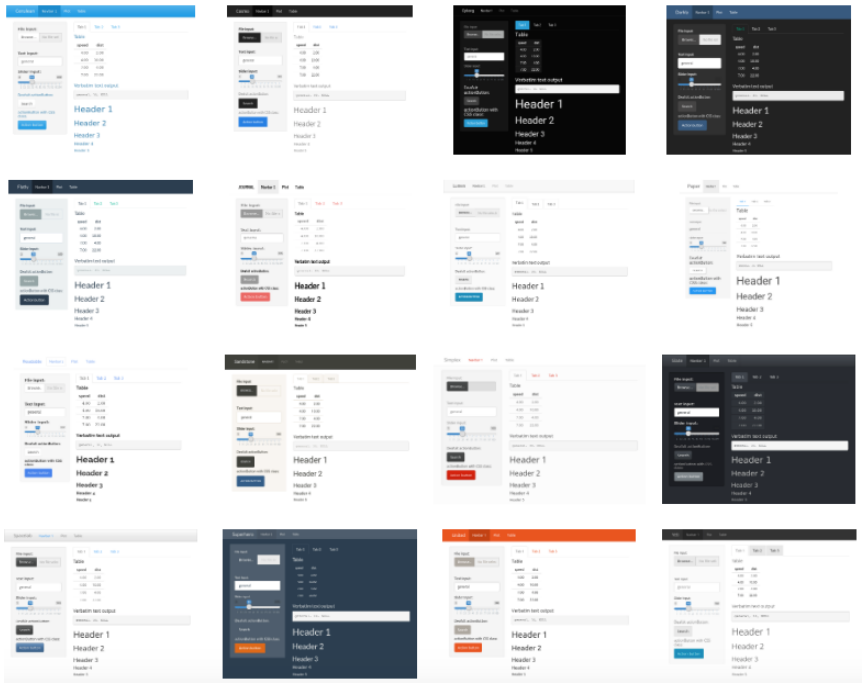
Geyser Eruption Duration

```{r}
renderPlot({
  hist(faithful$eruptions, probability = TRUE, breaks = as.numeric(input$n_breaks),
       xlab = "Duration (minutes)", main = "Geyser Eruption Duration")

  dens <- density(faithful$eruptions, adjust = input$bw_adjust)
  lines(dens, col = "blue")
})
```
```

(https://beta.rstudioconnect.com/jjallaire/shin
eruptions/)

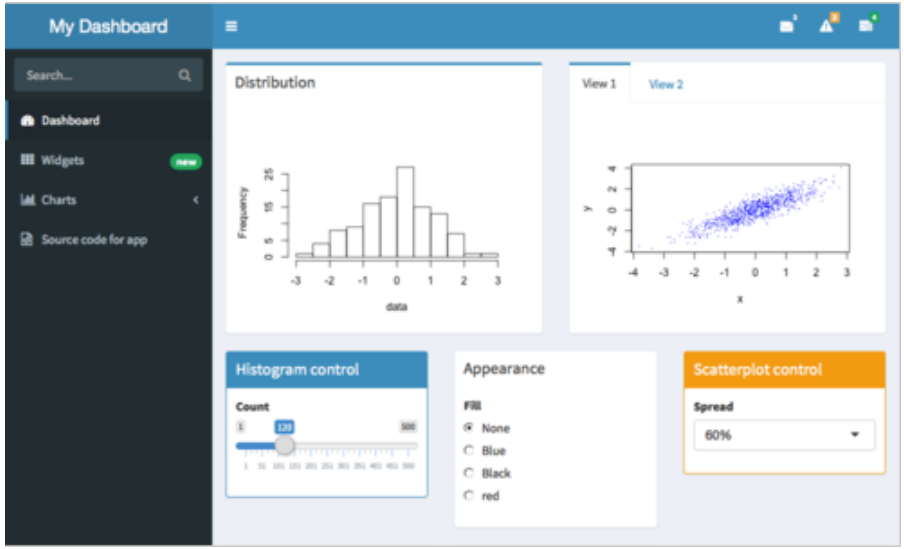
shinythemes (https://rstudio.github.io/shinythemes/)



```
## app.R ##
library(shinythemes)

shinyApp(
  ui = fluidPage(theme = shinytheme("united"),
    ...
  ),
  server = function(input, output) { }
)
```

shinydashboard (https://rstudio.github.io/shinydashboard/index.html)




```
## app.R ##
library(shiny)
library(shinydashboard)
```

shinyjqui (https://yang-tang.github.io/shinyjqui/)

Interaction funcions

There are five kinds of mouse interactions in jQuery UI library:

- **Draggable:** Allow elements to be moved using the mouse.
- **Droppable:** Create targets for draggable elements.
- **Resizable:** Change the size of an element using the mouse.
- **Selectable:** Use the mouse to select elements, individually or in a group.
- **Sortable:** Reorder elements in a list or grid using the mouse.

Here are the corresponding R wrappers in `shinyjqui`:

Functions	Description	Where_to_use
jqui_draggable	Enable or disable element's draggable interaction.	server
jqui_draggabled	Initialize an element as draggable.	ui
jqui_droppable	Enable or disable element's droppable interaction.	server
jqui_droppabled	Initialize an element as droppable.	ui
jqui_resizable	Enable or disable element's resizable interaction.	server
jqui_resizabled	Initialize an element as resizable.	ui
jqui_selectable	Enable or disable element's selectable interaction.	server
jqui_selectabled	Initialize an element as selectable.	ui
jqui_sortable	Enable or disable element's sortable interaction.	server
jqui_sortabled	Initialize an element as sortable.	ui

htmlwidgets (http://www.htmlwidgets.org/)

85 widgets registrados actualmente ( ):

- : Mapeado geoespacial interactivo
- : Creación de gráficos interactivos
- : Graficado de series temporales
- : Visualización gráfica de datos con D3 (https://d3js.org/)
- : Visualización de datos tabulares
- : Mapas de calor
- : Grafos y diagramas de flujo
- : Renderiza escenas creadas con rgl (http://rgl.neoscientists.org/about.shtml)

Leaflet (https://rstudio.github.io/leaflet/): mapas interactivos

- Llamadas en Shiny: y
- Tiles (http://leaflet-extras.github.io/leaflet-providers/preview/index.html)

```
library(leaflet)
leaflet() %>% addTiles() %>% addMarkers(lng=-0.3531, lat=39.4815, popup="FISABIO")
```

+
-

Plotly (https://plot.ly/r/): Navaja suiza de gráficos interactivos

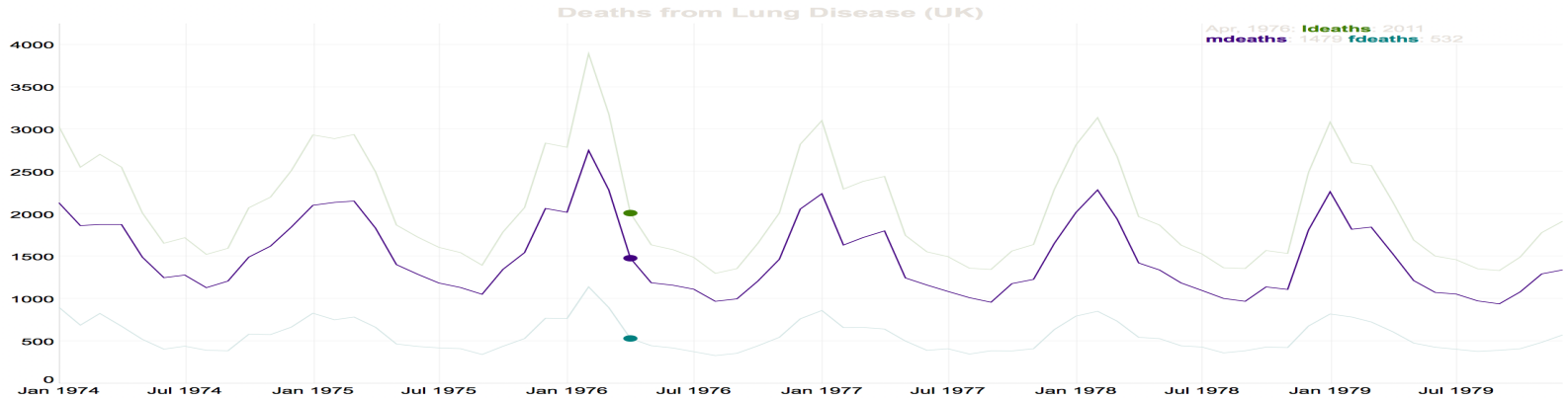
- Llamadas en Shiny: y

```
library(ggplot2, plotly)
p <- ggplot(data = diamonds, aes(x = cut, fill = clarity)) + geom_bar(position = "dodge")
ggplotly(p)
```

dygraphs (https://rstudio.github.io/dygraphs/): Gráficos interactivos para series temporales

- Llamadas en Shiny: y

```
library(dygraphs)
lungDeaths <- cbind(ldeaths, mdeaths, fdeaths)
dygraph(lungDeaths, main = "Deaths from Lung Disease (UK)") %>%
  dyHighlight(highlightCircleSize = 5,
              highlightSeriesBackgroundAlpha = 0.2,
              hideOnMouseOut = FALSE)
```



networkD3 (http://christophergandrud.github.io/networkD3/): Grafos interactivos

- Llamadas en Shiny: y

```
library(networkD3)
data(MisLinks, MisNodes)
forceNetwork(Links = MisLinks, Nodes = MisNodes, Source = "source", Target = "target", Value = "value", NodeID = '
```

DataTable (http://rstudio.github.io/DT/), paquete DT

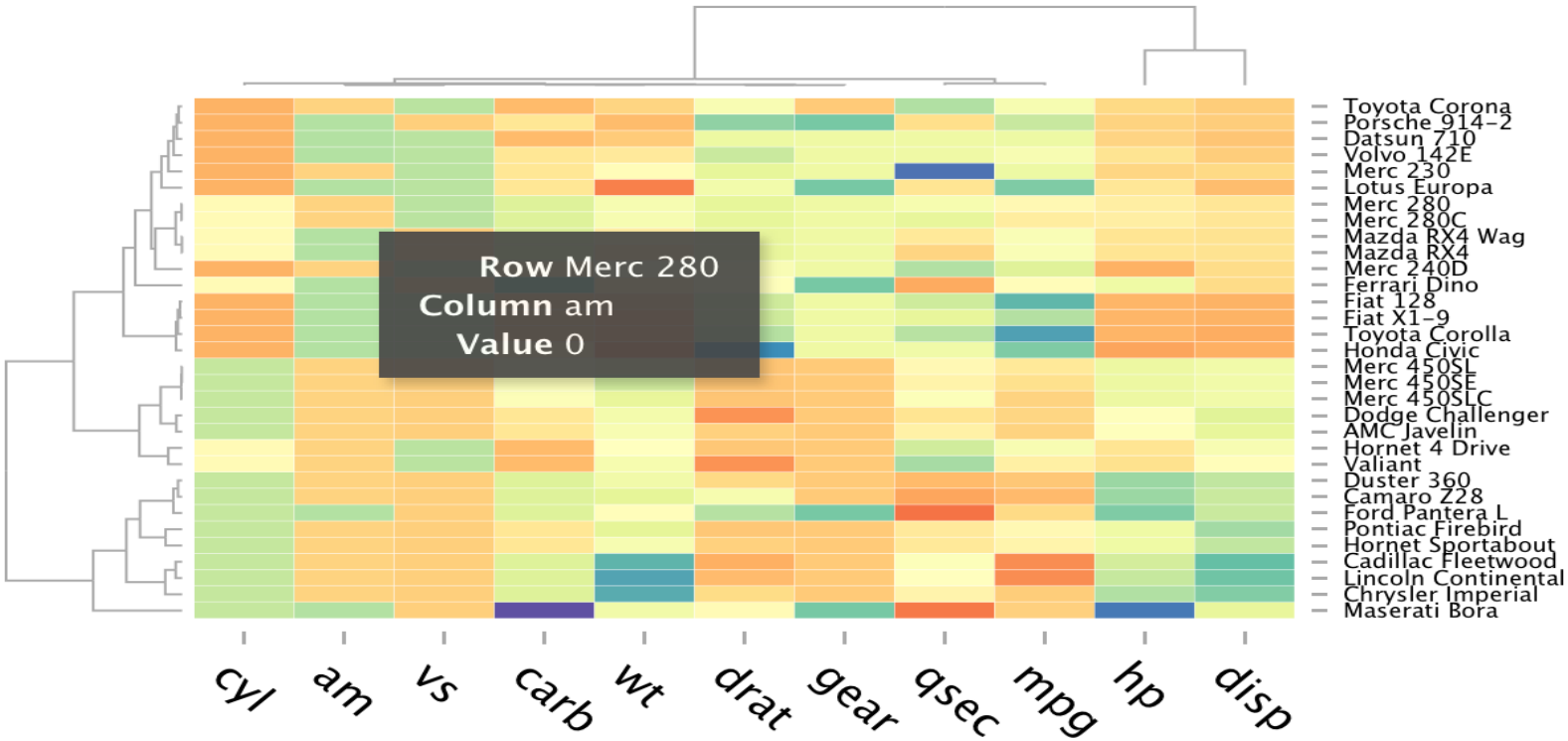
- Llamadas en Shiny: y

```
DT::datatable(iris, options = list(pageLength = 3, dom = "pt", language = list(url = '//cdn.datatables.net/plug-ir
```

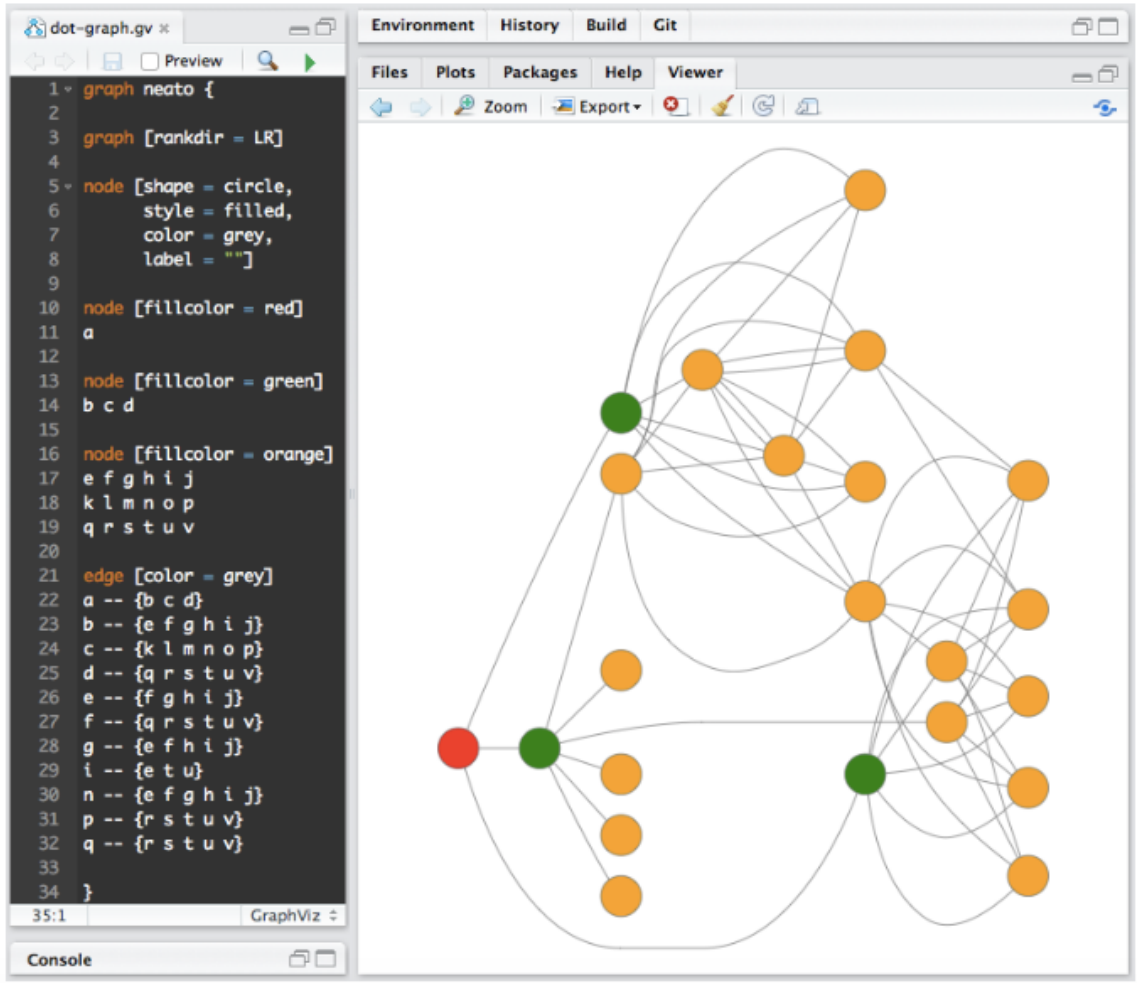
d3heatmap (https://github.com/rstudio/d3heatmap)

- Llamadas en Shiny: y

```
library(d3heatmap)
d3heatmap(mtcars, scale = "column", colors = "Spectral")
```



DiagrammeR (http://rich-iannone.github.io/DiagrammeR/)

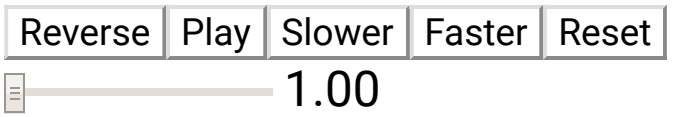


rglwidget (http://www.htmlwidgets.org/showcase_rglwidget.html)

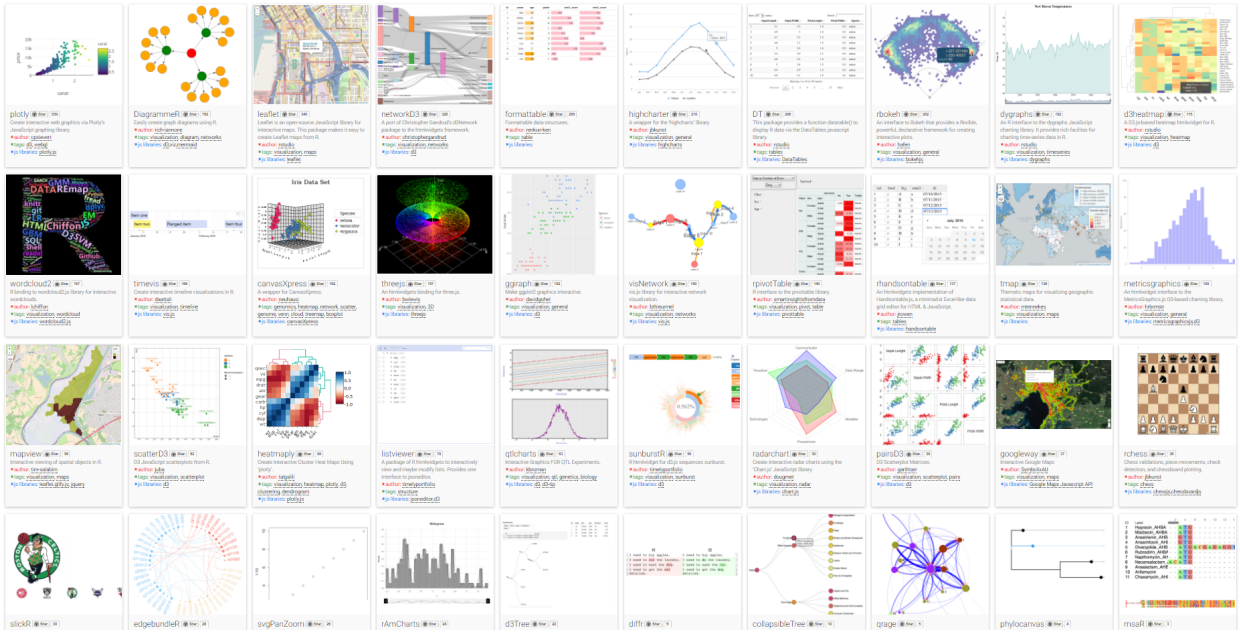
```
library(rgl, rglwidget, htmltools)

theta <- seq(0, 6*pi, len=100)
xyz <- cbind(sin(theta), cos(theta), theta)
lineid <- plot3d(xyz, type="l", alpha = 1:0, lwd =

browsable(tagList(
  rglwidget(elementId = "example", width = 500, hei
            controllers = "player"),
  playwidget("example",
             ageControl(births = theta, ages = c(0,
                        objids = lineid, alpha = c(
                        start = 1, stop = 6*pi, ste
                        rate = 6,elementId = "playe
```



y muchos más en http://gallery.htmlwidgets.org/ (http://gallery.htmlwidgets.org/)



3. Compartir las apps creadas con Shiny

3. Compartir las apps creadas con Shiny…

de forma local, con alguien que tiene R en su ordenador.

- , o

de forma global, con todo el mundo (sin necesidad de tener R).

- shinyapps.io
- Shiny Server
- RStudio Connect

Compartir apps Shiny

local

runUrl()

- Comprimir la carpeta de la app en un zip (https://es.wikipedia.org/wiki/Formato_de_compresi%C3%B3n_ZIP) y enlazar el archivo en una web

```
runUrl( "<link a la web>")
```

runGitHub()

- Alojar tu app en tu repositorio libre de GitHub (https://github.com/)

```
runGitHub( "<nombre de tu repositorio>", "<tu nombre de usuario>")
```

runGist()

- Alojar tu app en tu repositorio libre de GitHub (https://github.com/), manteniendo tu anonimato con Gist (https://gist.github.com/)
- Subir los archivos a Gist (https://gist.github.com/)

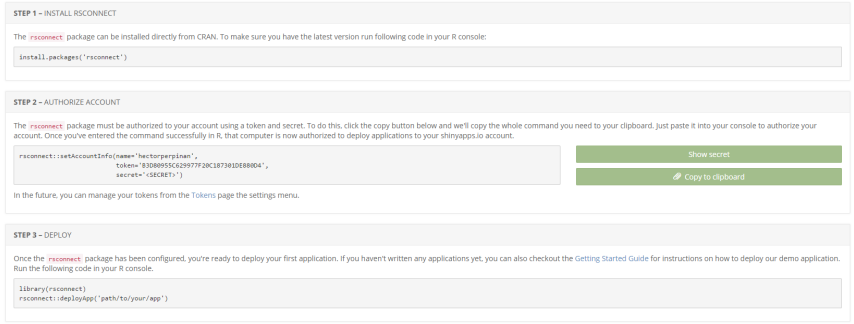
1. Gist (https://gist.github.com/) nos dará una Url
2. Los números finales de la Url son el **código gist**

```
runGist("código gist")
```

Compartir apps Shiny

global

shinyapps.io (https://www.shinyapps.io/)



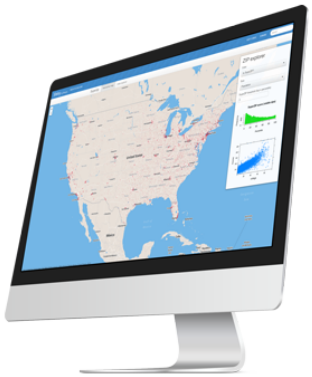
Shiny Server (https://www.rstudio.com/products/shiny/shiny-server/)

Put Shiny Web Apps Online

Shiny Server lets you put shiny web applications and interactive documents online. Take your Shiny apps and share them with your organization or the world.

Shiny Server lets you go beyond static charts, and lets you manipulate the data. Users can sort, filter, or change assumptions in real-time. Shiny server empower your users to customize your analysis for their specific needs and extract more insight from the data.

Shiny Server Pro adds enterprise grade scaling, security, and admin features to the basic open source edition.



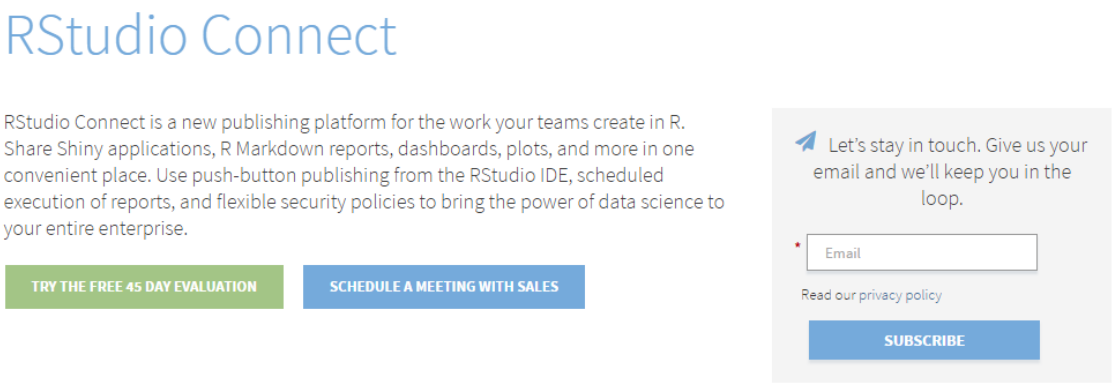
DOWNLOAD OPEN SOURCE

DOWNLOAD PRO

Description	Open Source	Pro
Overview		
Deploy Shiny applications and interactive documents to the internet	●	●
Move computation close to the data	●	●

RStudio Connect

(https://www.rstudio.com/products/connect/)



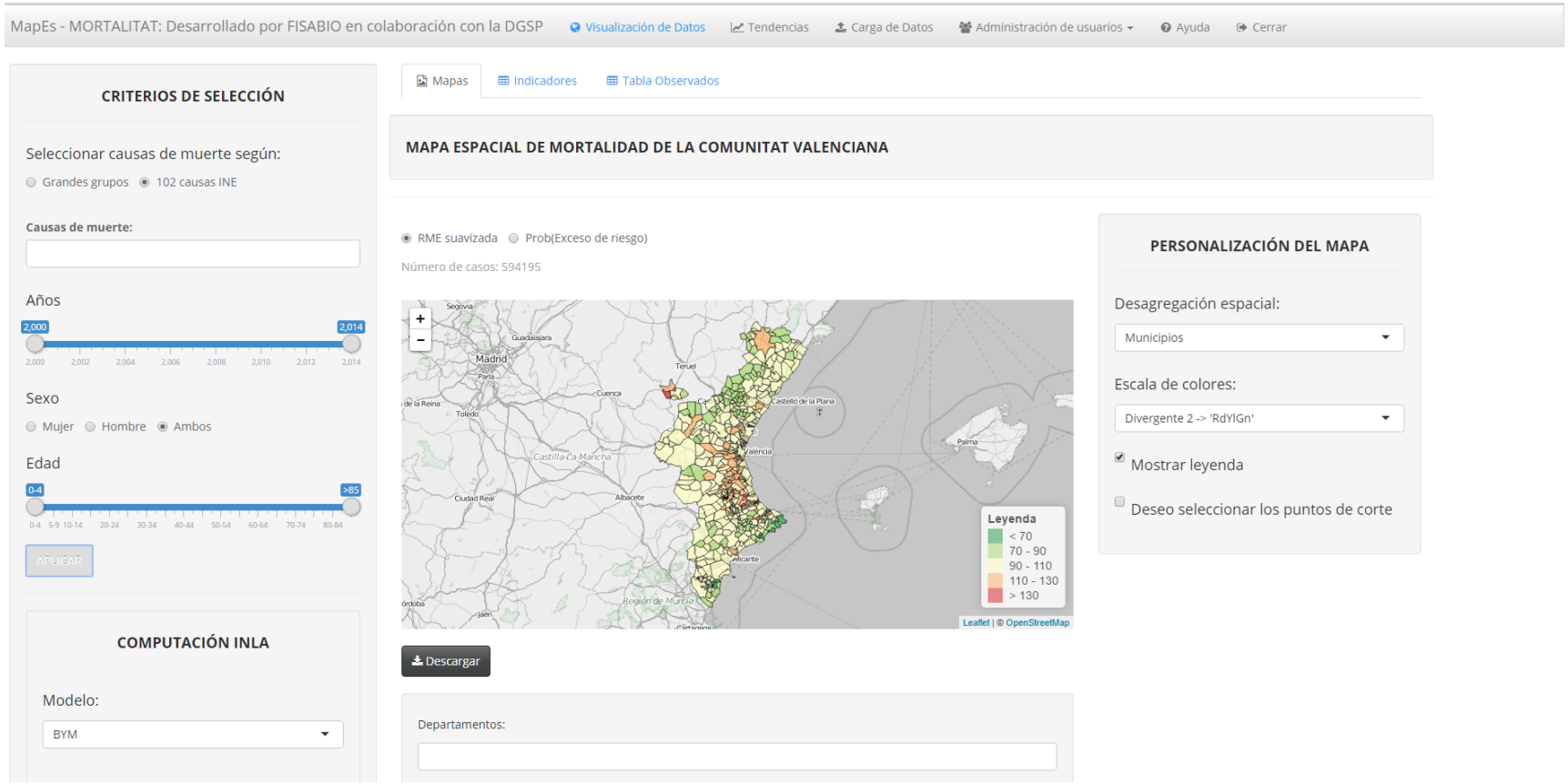
Comparativa características

Category	Description	RStudio Connect	Shiny Server Pro	Shinyapps.io
Overview	Commercial License (not AGPL)	●	●	●
	RStudio Support	●	●	●
	Deploy Shiny applications to the Web	●	●	●
	Push-button publishing from RStudio IDE	●		●
	One convenient place to share shiny apps, dashboards, R Markdown reports, and plots	●		
	Scheduled updates and distribution of reports	●		
	Self-managed content – view and manage what you've published or can access	●		Publishers Only
Security & Authentication	Password protect applications	●	●	●*
	Deploy Shiny applications behind firewalls	●	●	
	Controlled access via SSL and LDAP, Active Directory, Google OAuth, PAM, proxied authentication, or passwords	●	●	
Tuning & Scaling	Scale applications across multiple R processes	●	●	●
	Persistent R processes for faster load times	●		
Metrics & Management	Performance and resource metrics	●	●	●
	Health check endpoint	●	●	

* For shinyapps.io plans that include authentication, your application users must have a Google, Github or a shinyapps.io account

4. MapEs (una app desarrollada por FISABIO - DG Salud Pública)

MapEs (http://mapes.fisabio.san.gva.es/MapEs/)



Bibliografía y recursos

Leaflet Cheat Sheet (recursos/leaflet%20cheat%20sheet.pdf)

Shiny Cheat Sheet (https://www.rstudio.com/wp-content/uploads/2015/03/shiny-spanish.pdf)

Tutorial Shiny by RStudio (https://shiny.rstudio.com/tutorial/lesson1/)

RStudio Team. 2016. . Boston, MA: RStudio, Inc. http://www.rstudio.com/ (http://www.rstudio.com/).

Wickham, H. 2015. . Boca Raton, FL: CRC.