

# 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)
```

Título hecho con HTML

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)
```

Título hecho con R

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)
```

**Tamaño 1**

**Tamaño 2**

**Tamaño 3**

**Tamaño 4**

**Tamaño 5**

**Tamaño 6**

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)
```

**texto**

---

**texto**

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)
```

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

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)
```

Texto 1

Texto 2



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)
```

---

Párrafo 1

Párrafo 2

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  
)
```

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

```
# Definición del server  
server <- function(input, output) {}  
  
# App completa con los componentes ui y server  
shinyApp(ui, server)
```

# 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
  - **shinyjs**: 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

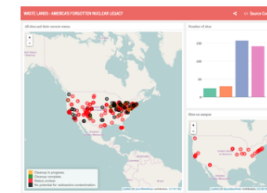
(<https://beta.rstudioconnect.com/jjallaire/showcase-storyboard/htmlwidgets-showcase-storyboard.html>)



MetricsGraphics: Tor Project



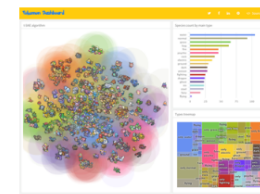
Shiny: kmeans clustering



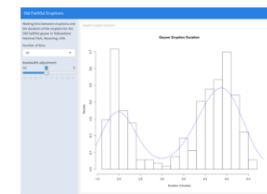
leaflet: nuclear waste sites



ggplotly: various examples



Pokemon characters with highcharter



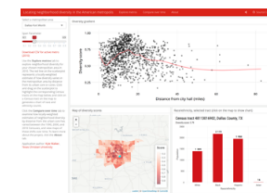
Shiny: Old faithful eruptions



Shiny: Diamonds explorer



Shiny: Embedding



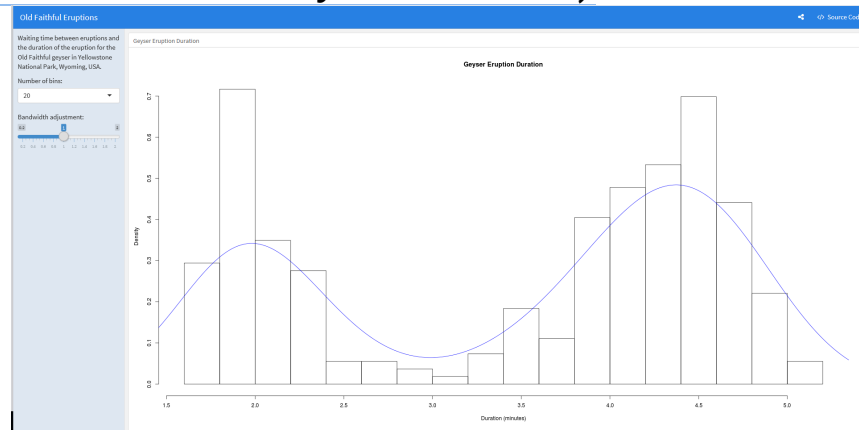
Shiny: Neighborhood diversity

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

# flexdashboard

(<http://rmarkdown.rstudio.com/flexdashboard/index.htm>)

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



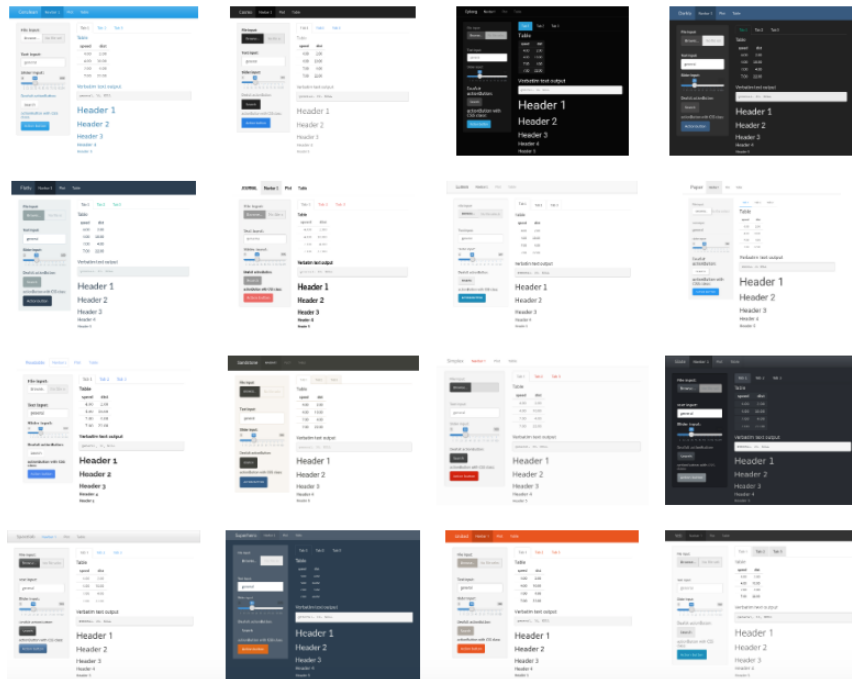
(<https://beta.rstudioconnect.com/jjallaire/showcase-storyboard/htmlwidgets-showcase-storyboard.html>)

(<https://beta.rstudioconnect.com/jjallaire/showcase-storyboard/eruptions/>)

```
1 ---
2 title: "Old Faithful Eruptions"
3 output:
4   flexdashboard::flex_dashboard:
5     orientation: rows
6     social: menu
7     source_code: embed
8 runtime: shiny
9 ---
10
11 {r global, include=FALSE}
12 # load data in 'global' chunk so it can be shared by all users of the dashboard
13 library(datasets)
14 data(faithful)
15
16
17 Column {.sidebar}
18 -----
19
20 Waiting time between eruptions and the duration of the eruption for the
21 Old Faithful geyser in Yellowstone National Park, Wyoming, USA.
22
23 {r}
24 selectInput("n_breaks", label = "Number of bins:",
25             choices = c(10, 20, 35, 50), selected = 20)
26
27 sliderInput("bw_adjust", label = "Bandwidth adjustment:",
28            min = 0.2, max = 2, value = 1, step = 0.2)
29
30
31 Column
32 -----
33
34 ## Geyser Eruption Duration
35
36 {r}
37 renderPlot({
38   hist(faithful$eruptions, probability = TRUE, breaks = as.numeric(input$n_breaks),
39        xlab = "Duration (minutes)", main = "Geyser Eruption Duration")
40
41   dens <- density(faithful$eruptions, adjust = input$bw_adjust)
42   lines(dens, col = "blue")
43 })
44
```

(<https://beta.rstudioconnect.com/jjallaire/showcase-storyboard/eruptions/>)

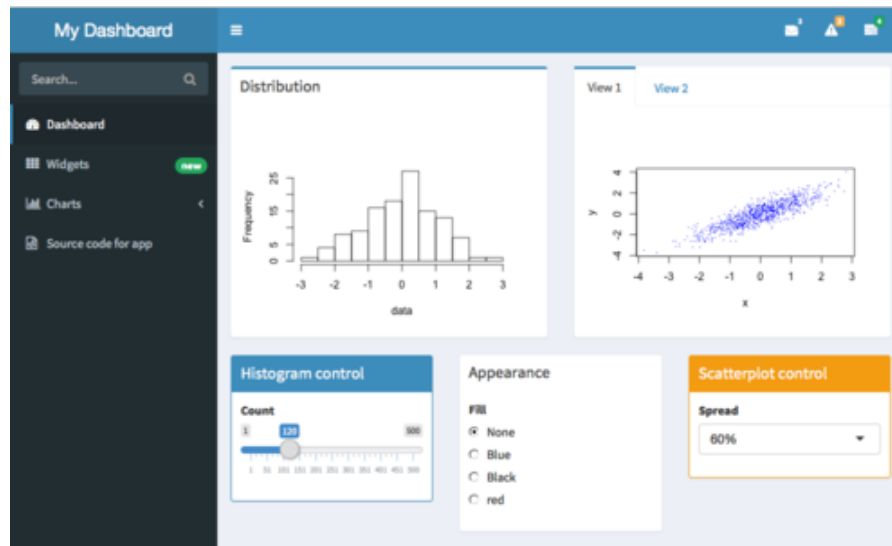
# shinythemes (<https://rstudio.github.io/shinythemes/>)



```
## 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)
```

# shinyjquery (<https://yang-tang.github.io/shinyjquery/>)

## Interaction functions

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 shinyjquery:

Functions	Description	Where_to_use
jquery_draggable	Enable or disable element's draggable interaction.	server
jquery_draggabled	Initialize an element as draggable.	ui
jquery_droppable	Enable or disable element's droppable interaction.	server
jquery_droppabled	Initialize an element as droppable.	ui
jquery_resizable	Enable or disable element's resizable interaction.	server
jquery_resizabled	Initialize an element as resizable.	ui
jquery_selectable	Enable or disable element's selectable interaction.	server
jquery_selectabled	Initialize an element as selectable.	ui
jquery_sortable	Enable or disable element's sortable interaction.	server
jquery_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: `leaflet` 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`

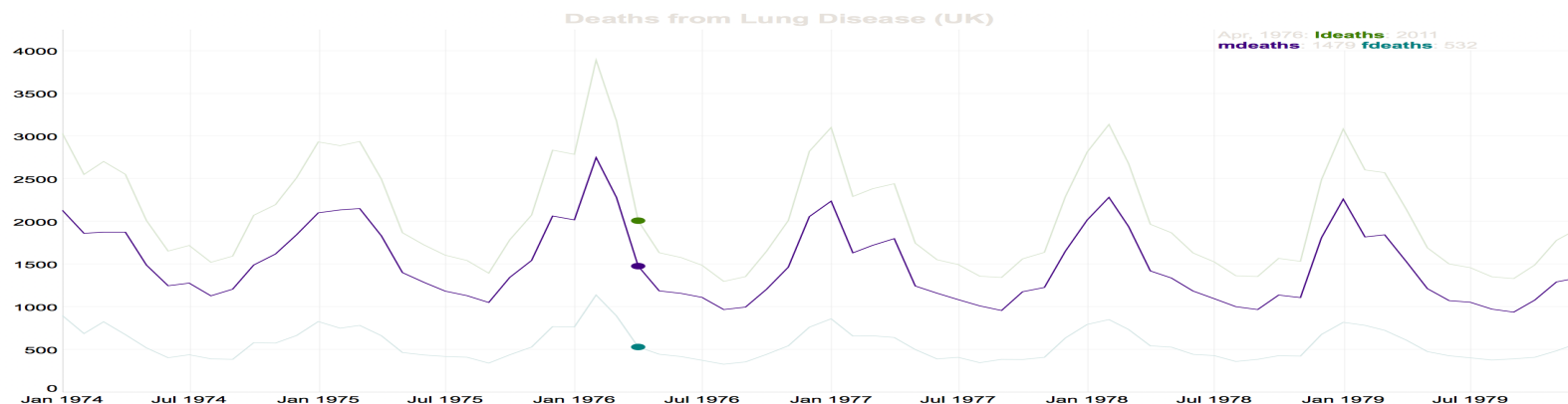
```
library(ggplot2, plotly)
p <- ggplot(data = diamonds, aes(x = cut, fill = clarity)) + geom_bar(position = "dodge")
ggplotly(p)
```



# 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**

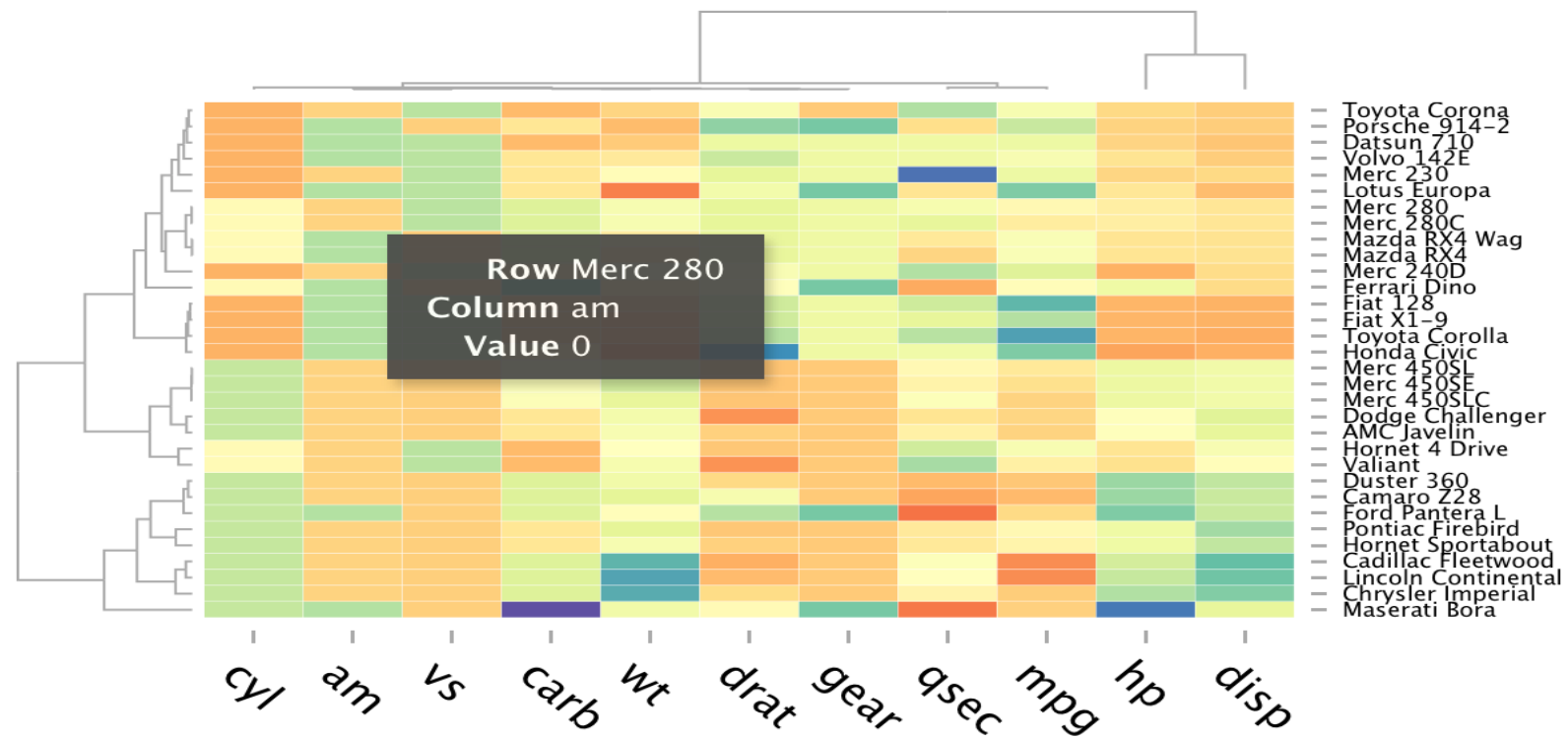
- Llamadas en Shiny: `y`

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

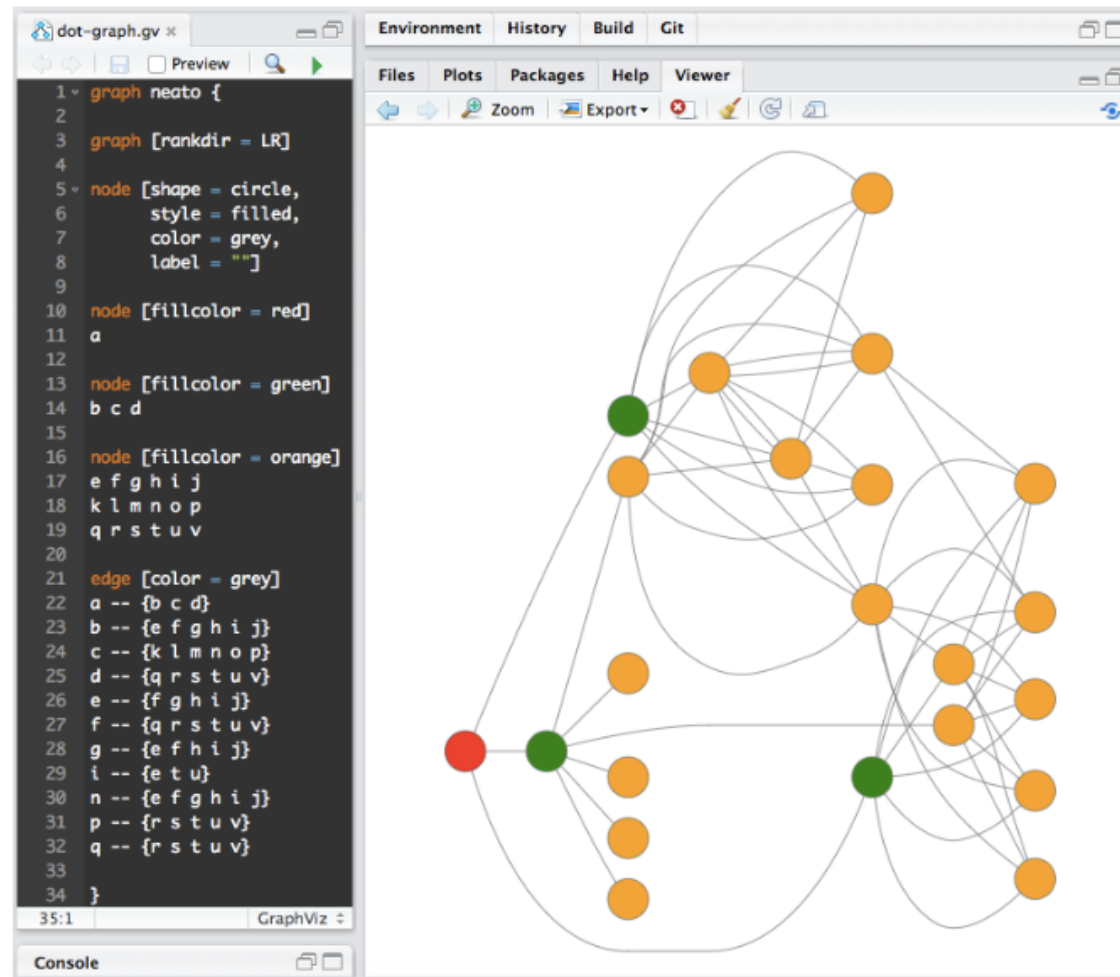
# d3heatmap (<https://github.com/rstudio/d3heatmap>)

- 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](http://www.htmlwidgets.org/showcase_rglwidget.html))

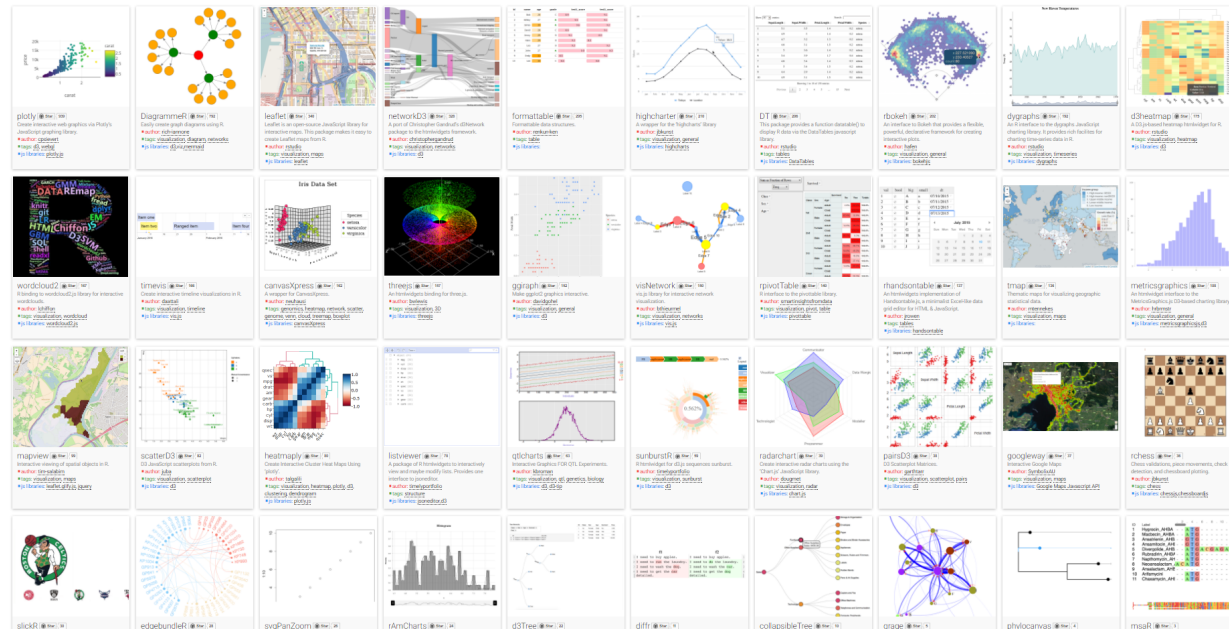
```
library(rgl, rglwidget, htmltools)

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.

- [RStudio Desktop](#), [RStudio Server](#) o [RStudio Cloud](#)

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](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()

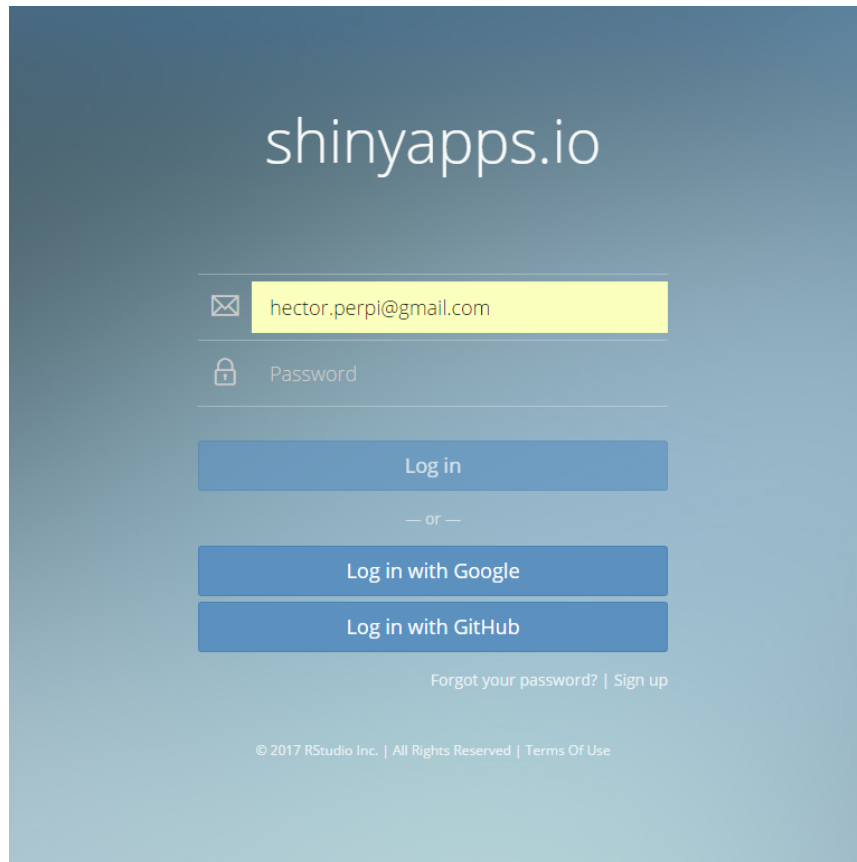
- Alojarse tu app en tu repositorio libre de GitHub (<https://github.com/>), manteniendo tu anonimato con Gist (<https://gist.github.com/>)
  - Subir los archivos a Gist (<https://gist.github.com/>)
1. [Gist \(https://gist.github.com/\)](https://gist.github.com/) nos dará una Url
  2. Los números finales de la Url son el **código gist**

```
runGist("código gist")
```

# **Compartir apps Shiny**

global

# shinyapps.io (<https://www.shinyapps.io/>)



The image shows the login page of shinyapps.io. It has a dark blue background with the shinyapps.io logo at the top. Below the logo, there are two input fields: one for email (containing 'hector.perpi@gmail.com') and one for password (containing 'Password'). Below these fields are three buttons: 'Log in', 'Log in with Google', and 'Log in with GitHub'. At the bottom, there is a link 'Forgot your password? | Sign up' and a copyright notice '© 2017 RStudio Inc. | All Rights Reserved | Terms Of Use'.

shinyapps.io

✉ hector.perpi@gmail.com

🔒 Password

Log in

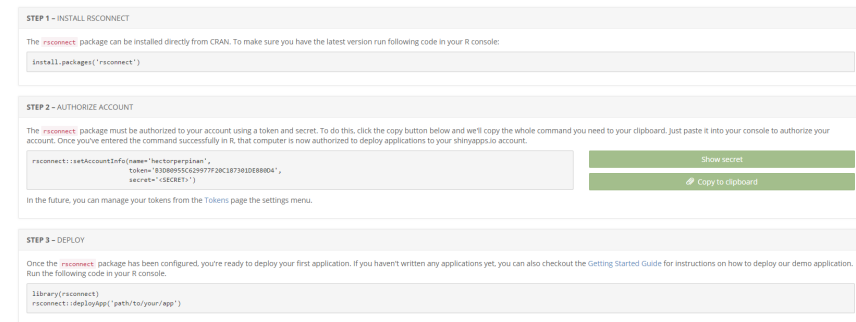
— or —

Log in with Google

Log in with GitHub

[Forgot your password? | Sign up](#)

© 2017 RStudio Inc. | All Rights Reserved | [Terms Of Use](#)



The image shows the setup steps for shinyapps.io. It is divided into three sections: STEP 1 - INSTALL RCONNECT, STEP 2 - AUTHORIZE ACCOUNT, and STEP 3 - DEPLOY. STEP 1 shows the command to install the rconnect package. STEP 2 shows the command to set the account info, with a 'Show secret' button and a 'Copy to clipboard' button. STEP 3 shows the command to deploy the application.

**STEP 1 - INSTALL RCONNECT**

The `rconnect` package can be installed directly from CRAN. To make sure you have the latest version run following code in your R console:

```
install.packages("rconnect")
```

**STEP 2 - AUTHORIZE ACCOUNT**

The `rconnect` package must be authorized to your account using a token and secret. To do this, click the copy button below and we'll copy the whole command you need to your clipboard. Just paste it into your console to authorize your account. Once you've entered the command successfully in R, that computer is now authorized to deploy applications to your shinyapps.io account.

```
rconnect::setAccountInfo(name="hectorperpi@",  
  token="83080955c29977f20c187301d88004",  
  secret="{SECRET}")
```

Show secret

📄 Copy to clipboard

In the future, you can manage your tokens from the Tokens page the settings menu.

**STEP 3 - DEPLOY**

Once the `rconnect` package has been configured, you're ready to deploy your first application. If you haven't written any applications yet, you can also checkout the [Getting Started Guide](#) for instructions on how to deploy our demo application. Run the following code in your R console.

```
library(rconnect)  
rconnect::deployApp("path/to/your/app")
```

# 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/>)

## RStudio Connect

RStudio Connect is a new publishing platform for the work your teams create in R. Share Shiny applications, R Markdown reports, dashboards, plots, and more in one convenient place. Use push-button publishing from the RStudio IDE, scheduled execution of reports, and flexible security policies to bring the power of data science to your entire enterprise.

TRY THE FREE 45 DAY EVALUATION

SCHEDULE A MEETING WITH SALES

Let's stay in touch. Give us your email and we'll keep you in the loop.

Email

[Read our privacy policy](#)

SUBSCRIBE

**INTRODUCING  
RSTUDIO CONNECT**

R

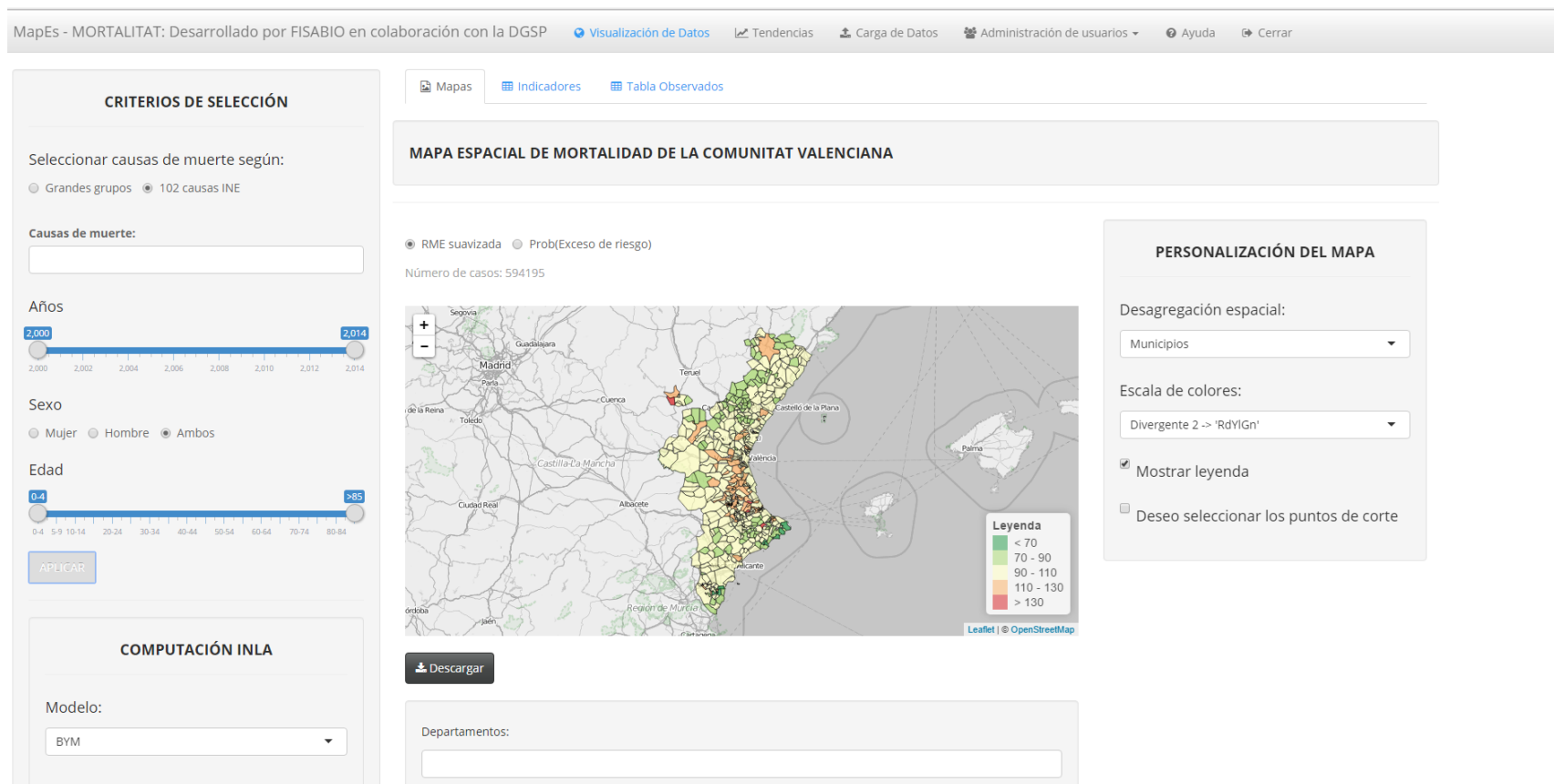
# Comparativa características

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 \(https://www.rstudio.com/wp-content/uploads/2015/03/shiny-spanish.pdf\)](https://www.rstudio.com/wp-content/uploads/2015/03/shiny-spanish.pdf)

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

RStudio Team. 2016.  
<http://www.rstudio.com/> (<http://www.rstudio.com/>).

. Boston, MA: RStudio, Inc.

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

