



Documentos Interactivos

Paloma Botella Rocamora - Servicio de Estudios Epidemiológicos y Estadísticas Sanitarias, CSISP-DGSP
01 de junio, 2017

Contenido

Contenido

- Introducción
- Documentos interactivos
- Crear un documento interactivo
- Widgets: Inputs & Outputs
- Ejemplos

Introducción

Introducción

Documentos interactivos

En la sesión anterior se ha introducido la creación de documentos mediante el entorno [\(R Markdown \(http://rmarkdown.rstudio.com/\)\)](http://rmarkdown.rstudio.com/)

Este entorno nos ofrece un gran abanico de posibilidades en la escritura (lenguaje) y la salida (Word, Pdf,...), pero en todos los casos produce **documentos estáticos**.

Los **documentos interactivos** permiten al **interactuar** con el documento modificando gráficos, tablas de resultados o cualquier contenido que haya sido diseñado en la creación del documento para poder ser modificado ([R Markdown-Documentos Interactivos \(http://rmarkdown.rstudio.com/lesson-14.html\)](http://rmarkdown.rstudio.com/lesson-14.html)).

Introducción

Reproducibilidad

¿Cómo ayudan este tipo de documentos en la reproducibilidad?

- Se trata de un tipo de documentos , por lo que las mismas ideas expuestas en la sesión de son aplicables a este tipo de documentos.
- Además, como mostraremos en esta sesión, la introducción de elementos que los hacen **dinámicos** proporciona gran potencia para resumir en un espacio reducido gran cantidad de información.
- Más allá de la reproducibilidad estricta, es una **herramienta útil** para **compartir** todos los resultados asociados a una investigación y que, por ejemplo, no han sido mostrados en un paper por disponer en general de espacio muy reducido.

Introducción

Utilidad

¿Por qué documentos interactivos?...

- **Réplicas** de análisis estadísticos de estimación de incidencia, estimación de tendencias, estimación de supervivencia, etc,...para diferentes combinaciones de grupos tumorales.
- **RMarkdown** me permitía programar un **análisis completo** y replicarlo para tantas combinaciones de grupos tumorales como deseara de forma muy eficiente. Además podía devolver un documento **Word** que mis compañeros epidemiólogos podían manejar perfectamente para publicar resultados.
- Pero...tal y como pensábamos en incorporar más y más resultados para cada grupo tumoral, el documento final crecía y crecía....llegando a tener **cerca de 1000 páginas!!!**

Estudio Supervivencia RTC - 2016

Periodo de incidencia: 2004-2012

Fecha fin de estudio: 31 de diciembre de 2013

Fuente: Registro de Tumores de Castellón,

Elaboración: Servicio de estudios epidemiológicos y estadísticas sanitarias. DGSP.

Se han calculado gráficas y tablas de supervivencia observada y relativa acumulada.

Software: Programa **R** (R Core Team, *R: A Language and Environment for Statistical Computing* [<https://www.R-project.org/>], 2015)

Estimación de la supervivencia relativa: función *rs.surv* de la librería *relsurv* (Computes an estimate of the relative survival curve using the Ederer II method, Pohar-Perme method). Es el mismo método que utiliza actualmente el programa **SEER** (<http://seer.cancer.gov/seerstat/tutorials/survival1/webprint/>), que indica en la página web en sus tutoriales *Starting with the November 2010 data submission, SEER uses the Ederer II method when calculating relative survival.*

Análisis descriptivo de algunos datos de interés para todos los tumores.....	21
Grupos tumorales y número de casos observados.....	21
Resumen de la supervivencia a 5 años de los grupos tumorales analizados.....	25
Extensiones según año de incidencia.....	28
Estadío según año de incidencia.....	29
Boca faringe (C00-C14).....	30
AMBOS SEXOS Boca faringe (C00-C14).....	30
Supervivencia observada y relativa.....	30
Análisis por EXTENSIÓN tumoral.....	32
Análisis por ESTADÍO tumoral.....	35
HOMBRES Boca faringe (C00-C14).....	36
Supervivencia observada y relativa.....	36
Análisis por EXTENSIÓN tumoral.....	38
Análisis por ESTADÍO tumoral.....	41
MUJERES Boca faringe (C00-C14).....	42
Supervivencia observada y relativa.....	42

Análisis por EXTENSIÓN tumoral.....	44
Análisis por ESTADÍO tumoral.....	47
Labio (C00).....	49
AMBOS SEXOS Labio (C00).....	49
Supervivencia observada y relativa.....	49
Análisis por EXTENSIÓN tumoral.....	51
Análisis por ESTADÍO tumoral.....	52
HOMBRES Labio (C00).....	53
Supervivencia observada y relativa.....	53
Análisis por EXTENSIÓN tumoral.....	55
Análisis por ESTADÍO tumoral.....	56
MUJERES Labio (C00).....	57
Supervivencia observada y relativa.....	57
Lengua (C01-C02).....	59
AMBOS SEXOS Lengua (C01-C02).....	59
Supervivencia observada y relativa.....	59
Análisis por EXTENSIÓN tumoral.....	61
Análisis por ESTADÍO tumoral.....	62
HOMBRES Lengua (C01-C02).....	63
Supervivencia observada y relativa.....	63
Análisis por EXTENSIÓN tumoral.....	65
Análisis por ESTADÍO tumoral.....	66
MUJERES Lengua (C01-C02).....	67
Supervivencia observada y relativa.....	67
Boca (C03-C06).....	69
AMBOS SEXOS Boca (C03-C06).....	69
Supervivencia observada y relativa.....	69
Análisis por EXTENSIÓN tumoral.....	71
Análisis por ESTADÍO tumoral.....	72
HOMBRES Boca (C03-C06).....	73
Supervivencia observada y relativa.....	73
MUJERES Boca (C03-C06).....	75
Supervivencia observada y relativa.....	75
Glándulas salivares (C07-C08).....	77

Primarios desconocidos (C80).....	796
AMBOS SEXOS Primarios desconocidos (C80)	796
Superervivencia observada y relativa.....	796
Análisis por EXTENSIÓN tumoral.....	798
Análisis por ESTADÍO tumoral.....	799
HOMBRES Primarios desconocidos (C80).....	800
Superervivencia observada y relativa.....	800
Análisis por EXTENSIÓN tumoral.....	802
Análisis por ESTADÍO tumoral.....	803
MUJERES Primarios desconocidos (C80).....	804
Superervivencia observada y relativa.....	804
Análisis por EXTENSIÓN tumoral.....	806
Análisis por ESTADÍO tumoral.....	807
Total tumores (excepto SNC y hemat) (C00-C43,C45-C69,C73-C80,C86-C90)	809
AMBOS SEXOS Total tumores (excepto SNC y hemat) (C00-C43,C45-C69,C73-C80,C86-C90).....	809
Superervivencia observada y relativa.....	809
Análisis por EXTENSIÓN tumoral.....	811
Análisis por ESTADÍO tumoral.....	814
Análisis por PERIODO de incidencia del tumor.....	815
HOMBRES Total tumores (excepto SNC y hemat) (C00-C43,C45-C69,C73-C80,C86-C90)	817
Superervivencia observada y relativa.....	817
Análisis por EXTENSIÓN tumoral.....	819
Análisis por ESTADÍO tumoral.....	822
Análisis por PERIODO de incidencia del tumor.....	823
MUJERES Total tumores (excepto SNC y hemat) (C00-C43,C45-C69,C73-C80,C86-C90).....	825
Superervivencia observada y relativa.....	825
Análisis por EXTENSIÓN tumoral.....	827
Análisis por ESTADÍO tumoral.....	830
Análisis por PERIODO de incidencia del tumor.....	831
Total tumores (C00-C96/C44).....	833
AMBOS SEXOS Total tumores (C00-C96/C44)	833
Superervivencia observada y relativa.....	833

Análisis por EXTENSIÓN tumoral.....	835
Análisis por ESTADÍO tumoral.....	838
Análisis por PERIODO de incidencia del tumor	841
HOMBRES Total tumores (C00-C96/C44).....	844
Superervivencia observada y relativa.....	844
Análisis por EXTENSIÓN tumoral.....	846
Análisis por ESTADÍO tumoral.....	849
Análisis por PERIODO de incidencia del tumor	852
MUJERES Total tumores (C00-C96/C44)	855
Superervivencia observada y relativa.....	855
Análisis por EXTENSIÓN tumoral.....	857
Análisis por ESTADÍO tumoral.....	860
Análisis por PERIODO de incidencia del tumor	863

Introducción

Utilidad

¿Por qué documentos interactivos?...

En ocasiones aplicamos **diferentes modelos** a **diferentes bancos de datos** y obtenemos **multitud de resultados** en cada ocasión....

Este tipo de herramientas permiten mostrar los resultados de forma que su consulta nos resulte útil en la exploración de los mismos.

Introducción

Utilidad

¿Por qué documentos interactivos?...

En un paper, por el espacio reducido habitualmente, realizamos una **selección pequeña de los resultados** que queremos mostrar, pero **disponemos de muchos más resultados** que pueden resultar interesantes para los lectores del trabajo. En función del tipo de resultados este tipo de documentos pueden servir para compartirlos de forma **sencilla, útil y atractiva**

Introducción

Documentos interactivos en RMarkdown

Existen dos posibles formas de crear un documento interactivo desde RMarkdown:

1. Utilizando elementos de **Shiny** (R Shiny (<https://shiny.rstudio.com/>)):
.
2. Incorporando **gráficos y/o tablas interactivas**, también conocidos como ([htmlwidgetsR \(http://www.htmlwidgets.org/\)](http://www.htmlwidgets.org/)) que existen en diferentes librerías de R:
.

Ambas formas pueden combinarse

Introducción

Shiny

Shiny es un **paquete** de RStudio que permite construir de manera muy fácil aplicaciones web interactivas con R.

<https://shiny.rstudio.com/> (<https://shiny.rstudio.com/>)



Documentos interactivos

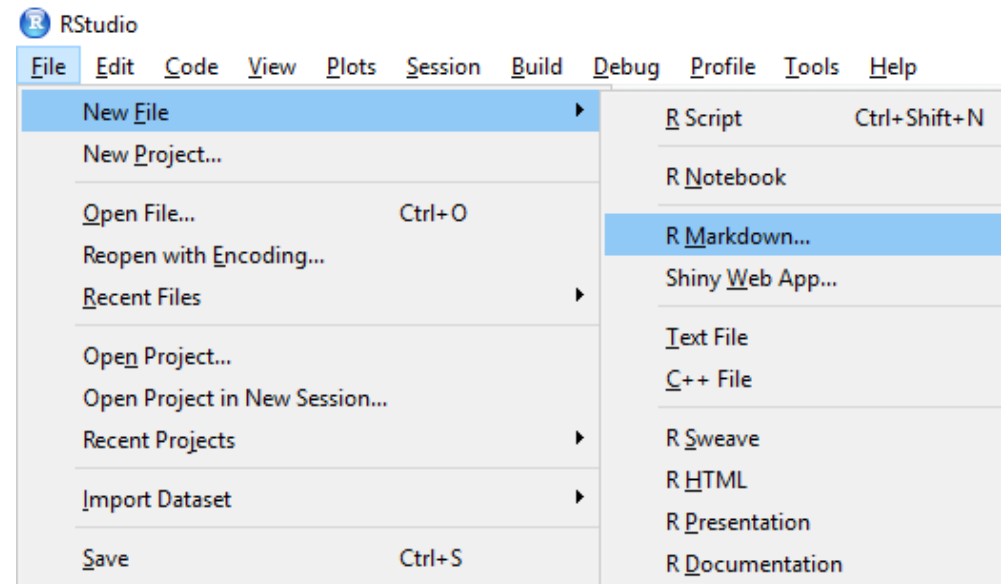
Documentos interactivos

Los **documentos interactivos** que vamos a introducir en esta sesión son **documentos** creados con **R Markdown** que permiten al lector/usuario, a través de **interactuar** con algunos objetos insertados en el mismo, visualizar diferentes resultados numéricos o gráficos en función de la selección realizada.

Veremos como crear documentos interactivos introduciendo en un documento R Markdown y vinculadas a los mismos.

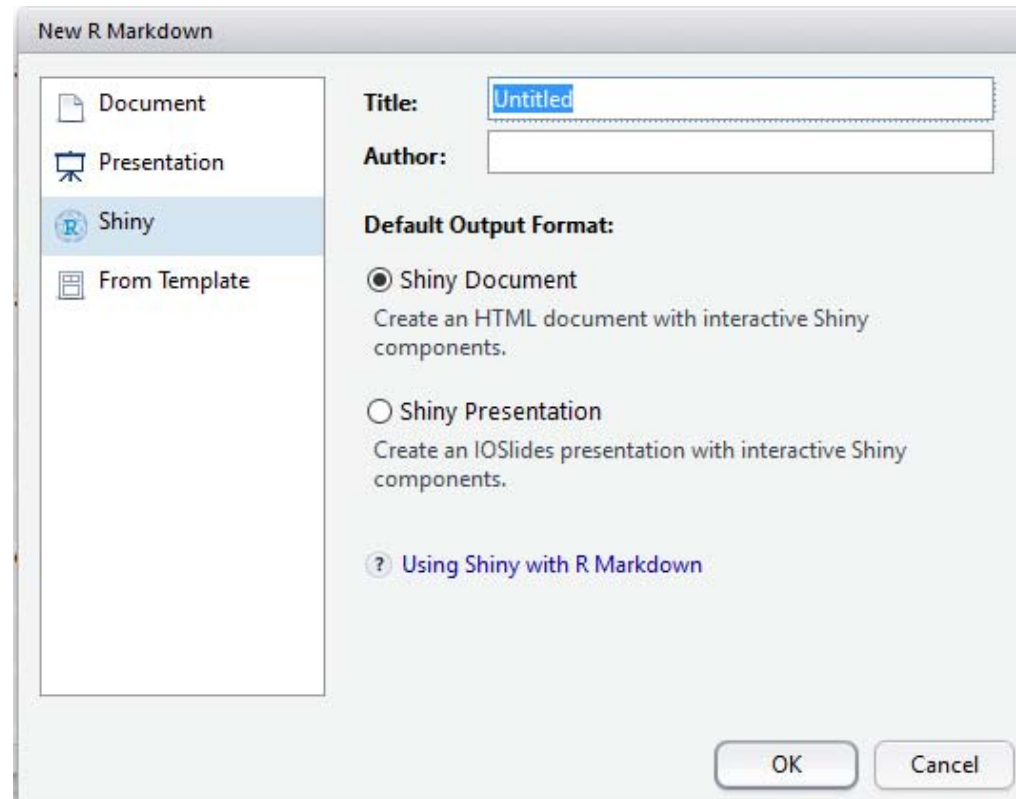
Cómo crear un documento interactivo

Paso 1



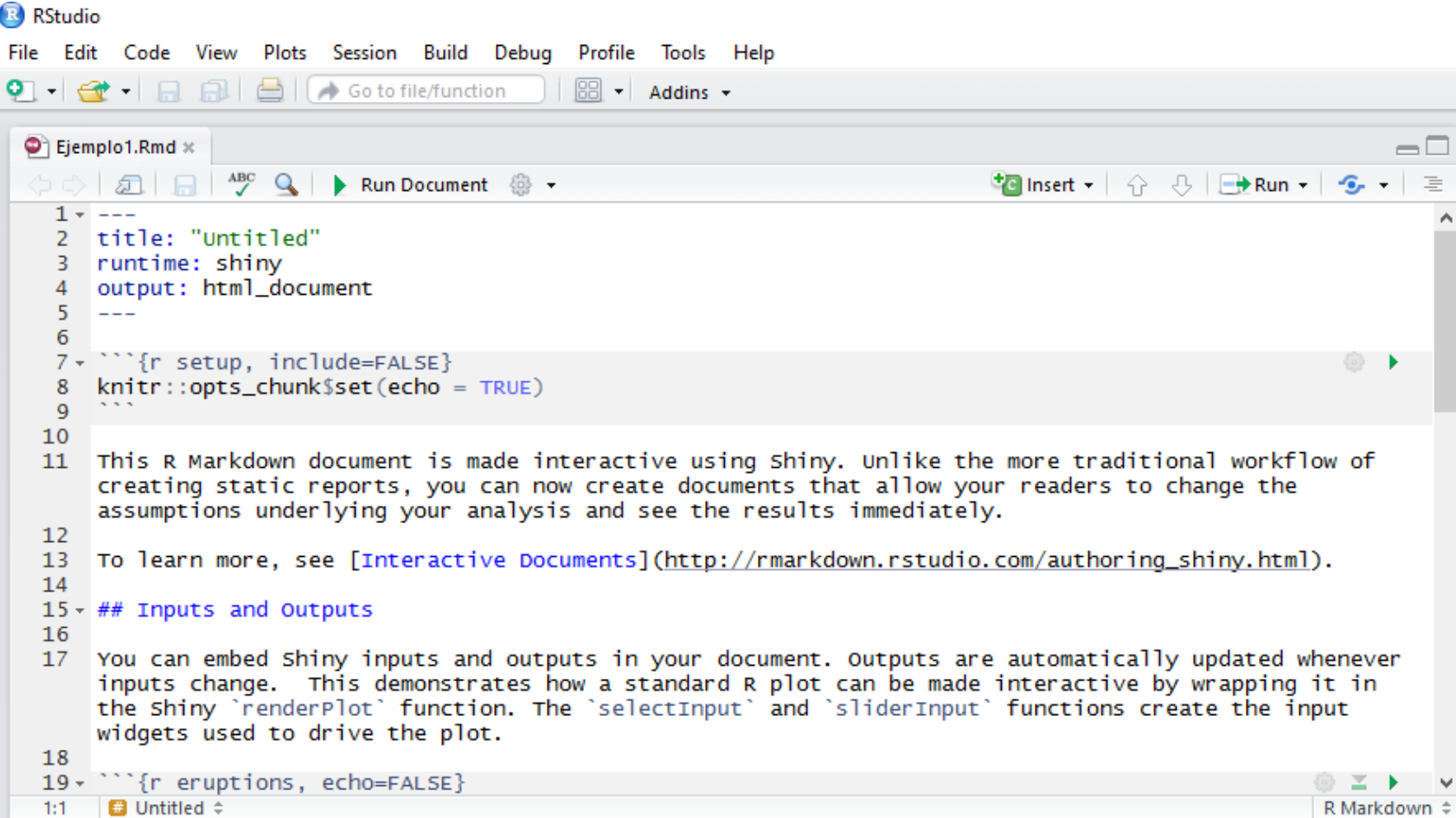
Cómo crear un documento interactivo

Paso 2



Documento interactivo

Ejemplo que genera RStudio (1)

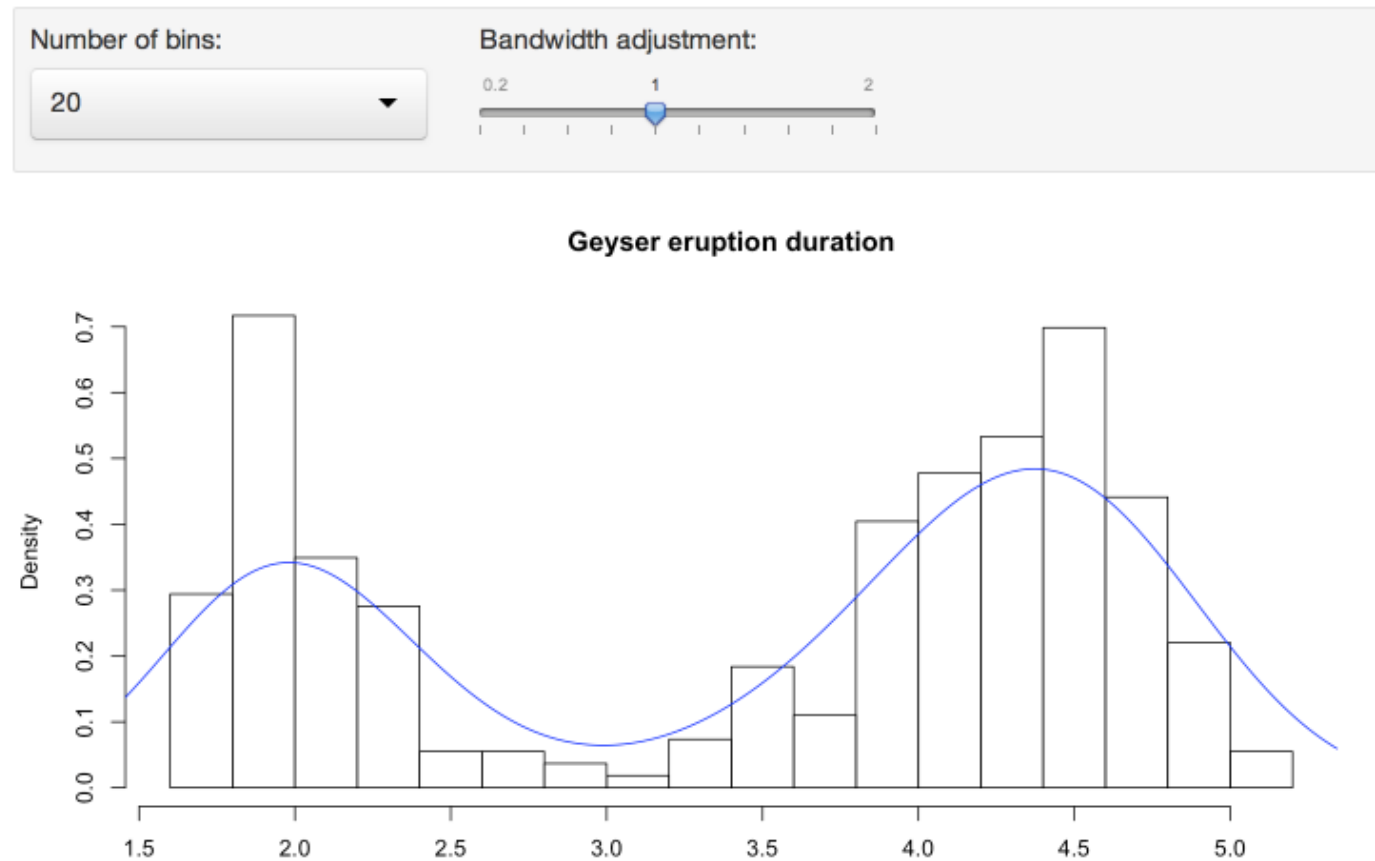


```
1 ---
2 title: "Untitled"
3 runtime: shiny
4 output: html_document
5 ---
6
7 ```{r setup, include=FALSE}
8 knitr::opts_chunk$set(echo = TRUE)
9 ```
10
11 This R Markdown document is made interactive using shiny. Unlike the more traditional workflow of
12 creating static reports, you can now create documents that allow your readers to change the
13 assumptions underlying your analysis and see the results immediately.
14
15 To learn more, see [Interactive Documents](http://rmarkdown.rstudio.com/authoring_shiny.html).
16
17 ## Inputs and Outputs
18
19 You can embed shiny inputs and outputs in your document. Outputs are automatically updated whenever
20 inputs change. This demonstrates how a standard R plot can be made interactive by wrapping it in
21 the shiny `renderPlot` function. The `selectInput` and `sliderInput` functions create the input
22 widgets used to drive the plot.
23
24 ```{r eruptions, echo=FALSE}
```

1:1 Untitled R Markdown

Documento interactivo

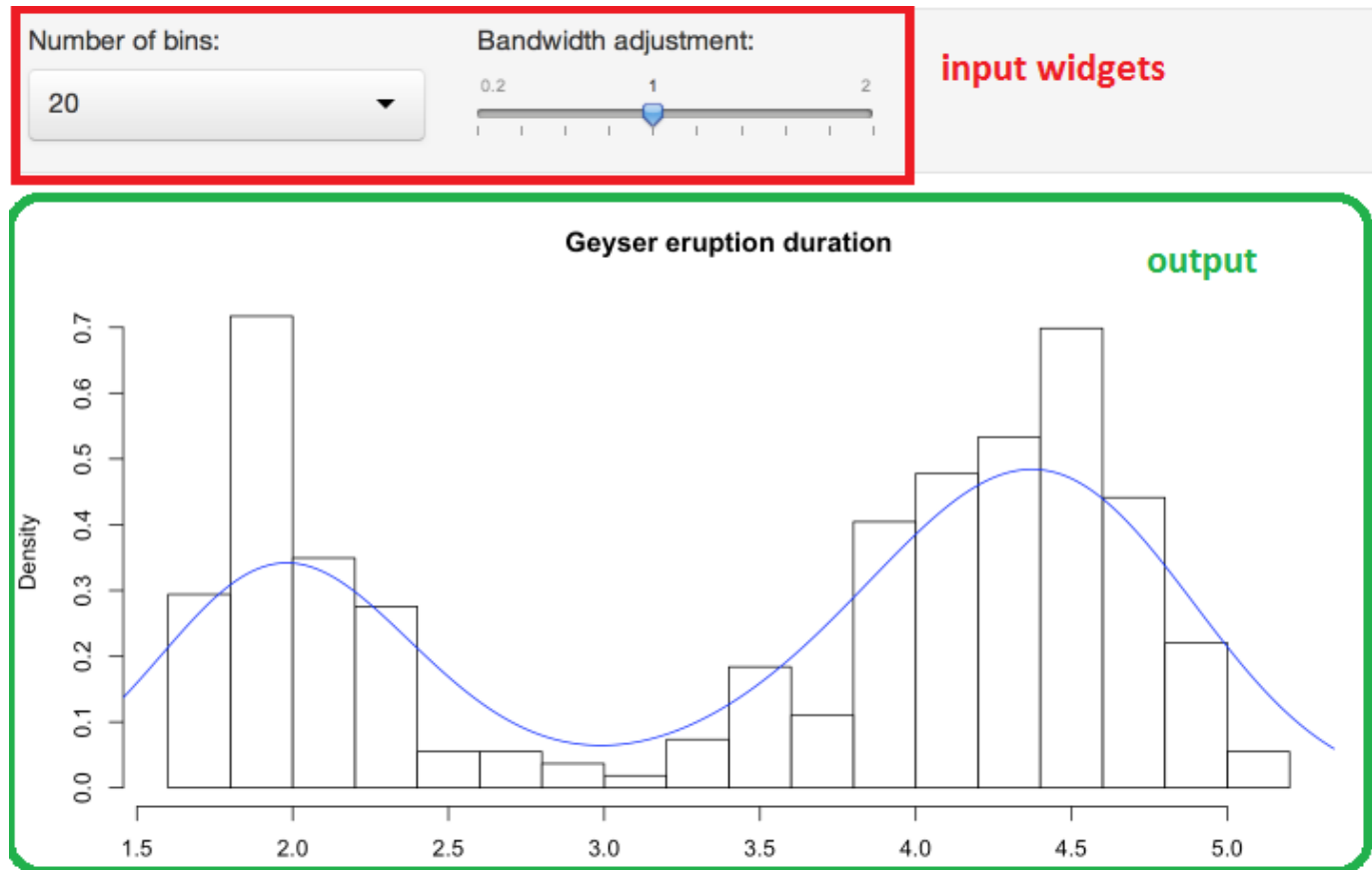
Ejemplo que genera RStudio (2)



Veamos directamente como funciona...Ejemplo 1

Documento interactivo

Ejemplo que genera RStudio (3)



Documento interactivo

Ejemplo que genera RStudio (4)

```
```{r eruptions, echo=FALSE}
inputPanel(
 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)
)

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")
})
```
```


Documento interactivo

Ejemplo que genera RStudio (5)

```
##{r eruptions, echo=FALSE}
inputPanel(
  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)
)

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")
})
```

Input

Output

Documento interactivo

Ejemplo que genera RStudio (5)

```
##{r eruptions, echo=FALSE}
inputPanel(
  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)
)

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")
})
```

Input

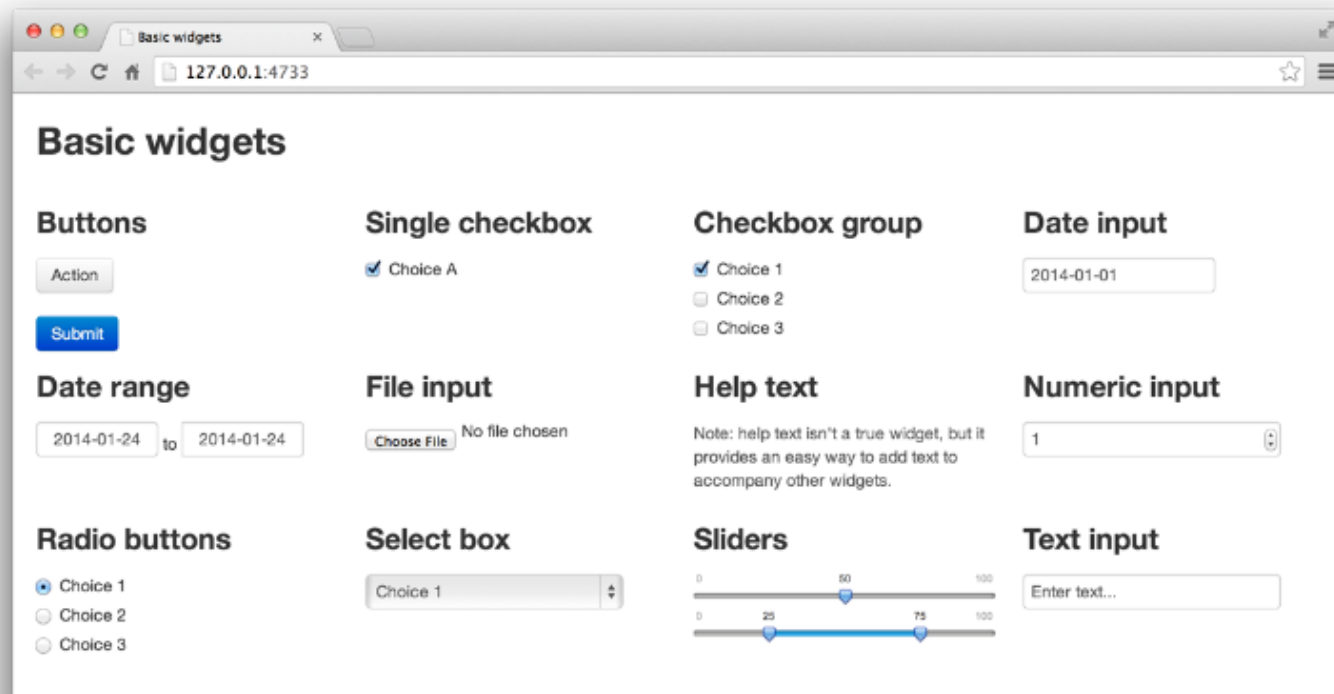
Output

Widgets: Inputs

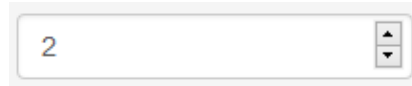
Widgets: Inputs

Se incluyen dentro del panel de :

Control widgets



Inputs: numericInput

A screenshot of a numeric input field. It is a light gray rectangular box with a thin border. Inside the box, the number '2' is displayed in a dark gray font. To the right of the text input area, there is a small square button with two arrows: an upward-pointing arrow on top and a downward-pointing arrow on the bottom.

- Permite introducir únicamente valores numéricos (si intentamos introducir cualquier otro símbolo tomará el valor NA)
- Permite teclear el número (con o sin decimales), aunque dispone de un selector que podemos utilizar.
- Su valor será el número introducido, en su defecto NA

Inputs: numericInput ()

```
numericInput(inputId, label, value, min = NA, max = NA, step = NA,width = NULL)
```

Arguments

inputId: the slot that will be used to access the value.

label: Display label for the control, or NULL for no label.

value: Initial value.

min: Minimum allowed value

max: Maximum allowed value

step: Interval to use when stepping between min and max

width: The width of the input, e.g. '400px', or '100%'; see `validateCssUnit`.

Inputs: numericInput ()

```
inputPanel(  
  numericInput("num", label = "Selecciona un valor", value = NULL,min=2,max=10,step=2)  
)  
  
renderPrint({  
  input$num  
})
```

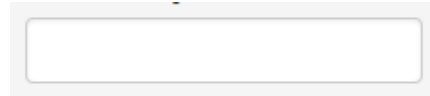
Selecciona un valor

[1] NA

Selecciona un valor

[1] 8

Inputs: `textInput`



- Permite introducir cualquier símbolo (numérico, carácter,...)
- Su valor será la cadena de caracteres introducida, en su defecto "".

Inputs: `textInput` ()

```
textInput(inputId, label, value = "", width = NULL, placeholder = NULL)
```

Arguments

placeholder: A character string giving the user a hint as to what can be entered into the control. Internet Explorer 8 and 9 do not support this option.

Aparece en gris claro el texto que introduzcamos en este parámetro como ayuda al usuario del documento interactivo. Aunque aparece dentro de la caja el valor del input correspondiente es "" mientras no se introduzca cualquier cadena de caracteres

Inputs: textInput ()

```
inputPanel(  
  textInput("titulo", label = "Título del gráfico", value = "",placeholder = "Introduce aquí el título"),  
  textInput("etiqx",labe="Título del eje x",value=""),  
  textInput("etiqy",labe="Título del eje y",value="")  
)  
  
renderPrint(input$titulo)  
  
renderPlot({  
  plot(1:10,1:10,main=input$titulo,xlab=input$etiqx, ylab=input$etiqy)  
})
```

Inputs: textInput ()

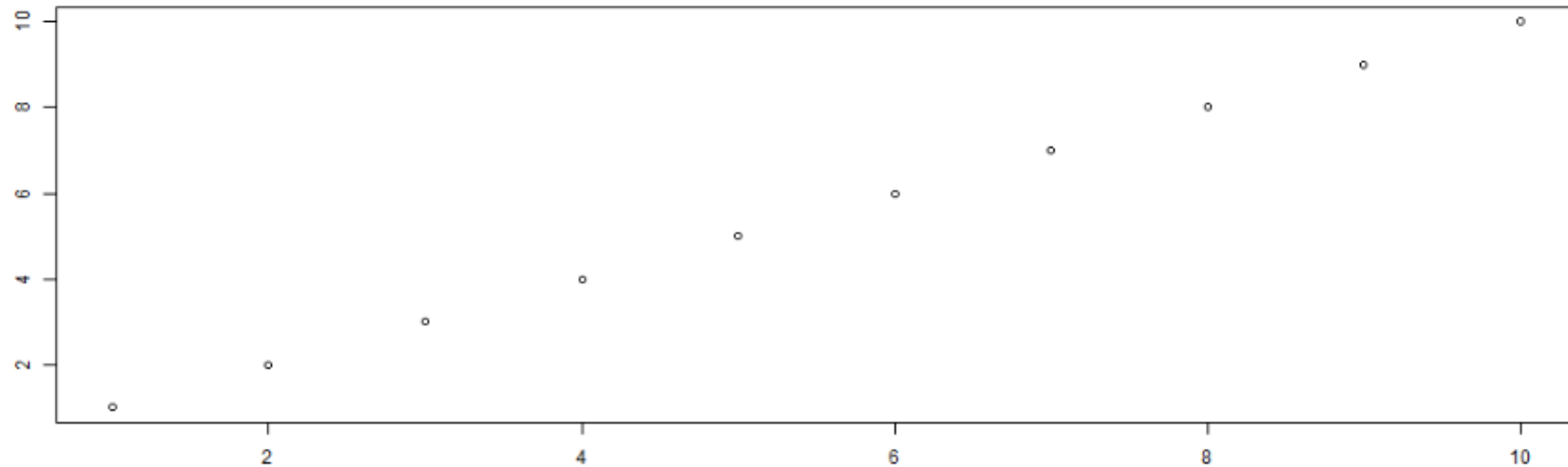
Título del gráfico

Introduce aquí el título

Título del eje x

Título del eje y

```
[1] ""
```



Inputs: textInput ()

Título del gráfico

Título para el gráfico

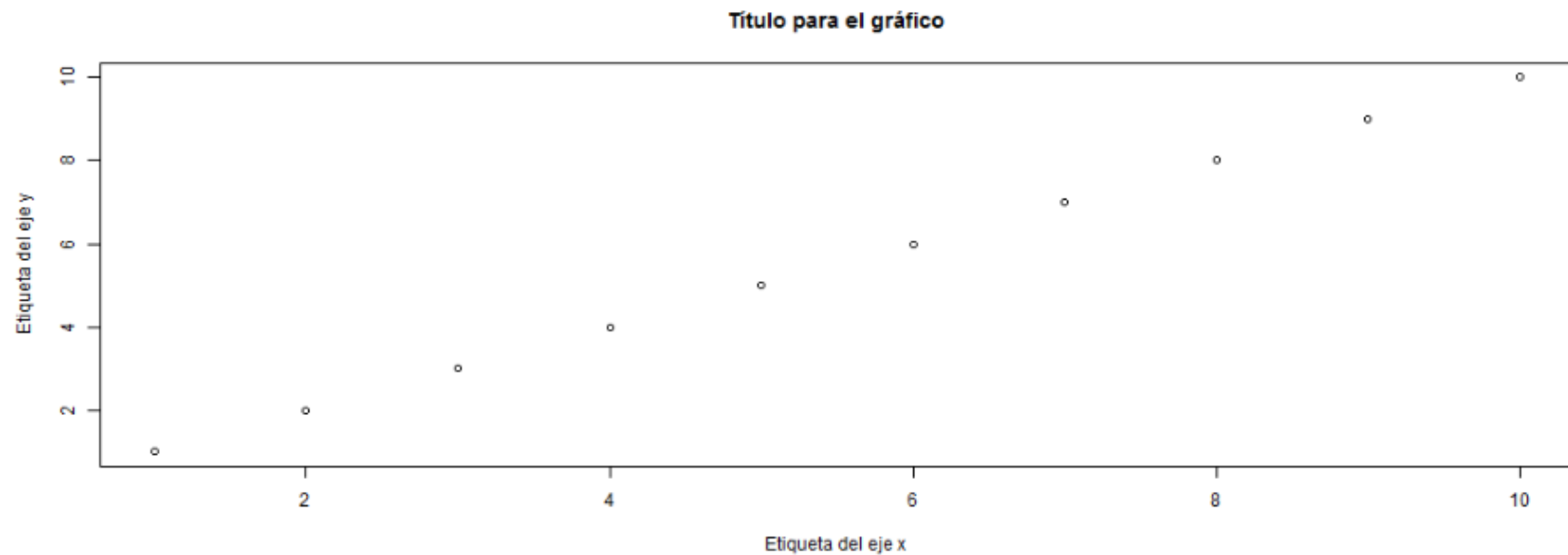
Título del eje x

Etiqueta del eje x

Título del eje y

Etiqueta del eje y

```
[1] "Título para el gráfico"
```



Inputs: dateInput/dateRangeInput



2017-01-01

2016-01-01 to 2016-12-31

- Permite introducir una fecha, o bien un rango de fechas.
- Ofrece diferentes formatos para la fecha (o fechas) y para la forma de selección y visualización.
- Su valor será la fecha introducida en el `value`, o bien un vector con dos elementos en el caso del `dateRangeInput`.
- Siempre toman un valor, si no se selecciona ninguno tomarán la fecha actual (la del sistema)
- En el caso del `dateRangeInput` no controla que la primera fecha sea menor que la segunda, recoge dos fechas independientes.

Inputs: `dateInput/dateRangeInput` ()

```
dateInput(inputId, label, value = NULL, min = NULL, max = NULL,  
  format = "yyyy-mm-dd", startview = "month", weekstart = 0,  
  language = "en", width = NULL)
```

Arguments

value: The starting date. Either a Date object, or a string in yyyy-mm-dd format. If NULL (the default), will use the current date in the client's time zone.

format: The format of the date to display in the browser. Defaults to "yyyy-mm-dd".

El valor inicial debemos darlo en el formato que indica, , aunque en función de los parámetros que veremos a continuación se visualizará de una forma u otra. Por ejemplo, podemos seleccionar ver las fechas de la siguiente forma . Se puede profundizar en el parámetro en la ayuda del control (visualizar el número de mes o el nombre, etc,...)

Inputs: dateInput/dateRangeInput ()

```
dateInput(inputId, label, value = NULL, min = NULL, max = NULL,  
  format = "yyyy-mm-dd", startview = "month", weekstart = 0,  
  language = "en", width = NULL)
```

Arguments

min: The minimum allowed date. Either a Date object, or a string in yyyy-mm-dd format.

max: The maximum allowed date. Either a Date object, or a string in yyyy-mm-dd format.

Este control ofrece una ventana para seleccionar la fecha deseada. Esta fecha mínima y máxima refleja las fechas que se pueden seleccionar en ese calendario. En ese calendario únicamente nos permitirá seleccionar fechas que se encuentren en el rango definido. Sin embargo, tecleando manualmente el control permite recoger cualquier fecha, incluso fuera del rango que indicamos en y .

Inputs: dateInput/dateRangeInput ()

```
dateInput(inputId, label, value = NULL, min = NULL, max = NULL,  
  format = "yyyy-mm-dd", startview = "month", weekstart = 0,  
  language = "en", width = NULL)
```

startview: The date range shown when the input object is first clicked. Can be "month" (the default), "year", or "decade".

En este parámetro se recoge únicamente la configuración del objeto calendario que se despliega para que podamos seleccionar en él una fecha. Por defecto aparece un mes (el de la fecha que contiene o el actual en otro caso), aunque podemos querer ver los meses del año en cuestión o bien los años de la década, y así ir seleccionando año y mes hasta visualizar los días del mes.

Inputs: dateInput/dateRangeInput ()

```
dateInput(inputId, label, value = NULL, min = NULL, max = NULL,  
  format = "yyyy-mm-dd", startview = "month", weekstart = 0,  
  language = "en", width = NULL)
```

Arguments

weekstart: Which day is the start of the week. Should be an integer from 0 (Sunday) to 6 (Saturday).

Podemos elegir `weekstart=1` si queremos visualizar las semanas comenzando en el lunes.

language: The language used for month and day names. Default is `"en"`. Other valid values include `"es"`, `"ca"`,

Elegiremos `language="es"` si queremos ver los nombres de los días y meses en español.

Inputs: dateInput/dateRangeInput ()

```
inputPanel(dateInput("Fecha", label = "Fecha de ingreso",  
                    value="2017-06-07",format="dd/mm/yyyy",  
                    language="es",weekstart = 1,  
                    min="2017-06-01",max="2017-06-15")  
)
```

```
renderPrint({ input$Fecha })
```

```
inputPanel(  
dateRangeInput("FechaDesdeHasta", label = "Fecha desde hasta",start="2016-01-01",end="2016-12-31")  
)
```

```
renderPrint({input$FechaDesdeHasta })
```

Inputs: dateInput/dateRangeInput ()

Fecha de ingreso

07/06/2017

[1] "2017-06-07"

Fecha desde hasta

2016-01-01 to 2016-12-31

[1] "2016-01-01" "2016-12-31"

Inputs: dateInput/dateRangeInput ()

Fecha de ingreso

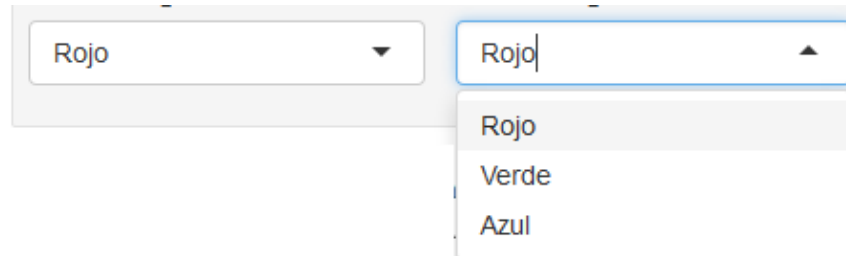
12/06/2017

Junio 2017

| Lu | Ma | Mi | Ju | Vi | Sa | Do |
|----|----|----|----|----|----|----|
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

31"

Inputs: selectInput



- Permite seleccionar únicamente valores definidos en el control
- Si intentamos introducir cualquier otro símbolo seleccionará el que tiene definido por defecto o el primero.
- Según se configure con los parámetros permitirá seleccionar un único valor o varios, y en este último caso su valor será un vector.

Inputs: selectInput ()

```
selectInput(inputId, label, choices, selected = NULL, multiple = FALSE,  
            selectize = TRUE, width = NULL, size = NULL)
```

Arguments

choices: List of values to select from. If elements of the list are named then that name rather than the value is displayed to the user.

Este parámetro recoge tanto las opciones que queremos que se puedan seleccionar como los valores que se guardarán internamente si son seleccionadas.

Inputs: selectInput ()

```
selectInput(inputId, label, choices, selected = NULL, multiple = FALSE,  
            selectize = TRUE, width = NULL, size = NULL)
```

Arguments

selected: The initially selected value (or multiple values if `multiple = TRUE`). If not specified then defaults to the first value for single-select lists and no values for multiple select lists.

En el parámetro `selected` se seleccionará, de las opciones posibles, la que estará marcada por defecto. Si no se indica esta opción, o en este parámetro se proporciona un valor no posible se seleccionará por defecto la primera de las opciones.

Inputs: selectInput ()

```
selectInput(inputId, label, choices, selected = NULL, multiple = FALSE,  
            selectize = TRUE, width = NULL, size = NULL)
```

Arguments

multiple: Is selection of multiple items allowed?

Parámetro que recoge si permitimos selección de varios ítems () o de un único ítem (). Si permitimos selección múltiple se recogerá un vector con todas las opciones.

Inputs: selectInput ()

```
inputPanel(  
  selectInput("ColorSel1", label = "Color del gráfico 1",  
    choices = list("Rojo" = 1, "Verde" = 2, "Azul" = 3),  
    selected = 1,multiple=TRUE),  
  selectInput("ColorSel2",label="Color del gráfico 2",  
    choices=list("Rojo"="red","Verde"="green","Azul"="blue"),  
    selected="Rojo")  
)  
  
renderPrint(input$ColorSel1)  
  
renderPrint(input$ColorSel2)  
  
renderPlot({  
  par(mfrow=c(1,2))  
  plot(density(rnorm(1000,0,1)),col=as.numeric(input$ColorSel1)+1,xlab="",ylab="",main="Gráfico 1")  
  plot(density(rnorm(1000,0,1)),col=input$ColorSel2,xlab="",ylab="",main="Gráfico 2")  
})
```

Inputs: selectInput ()

Color del gráfico 1

Rojo

Color del gráfico 2

Rojo

```
[1] "1"
```

```
[1] "red"
```

Gráfico 1

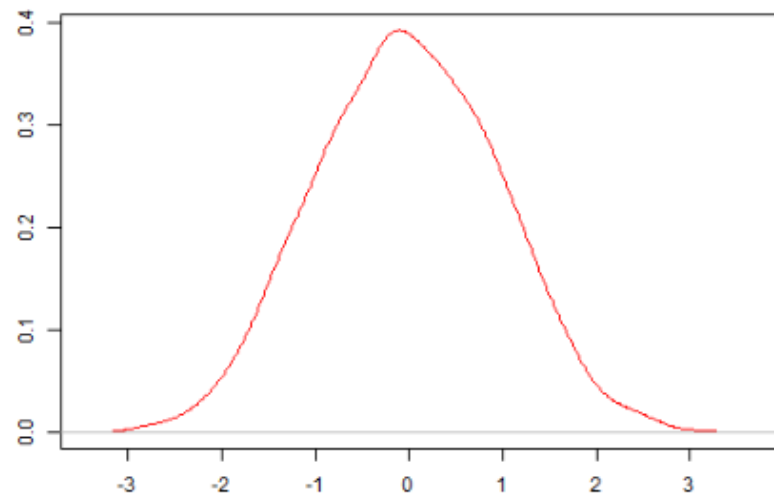
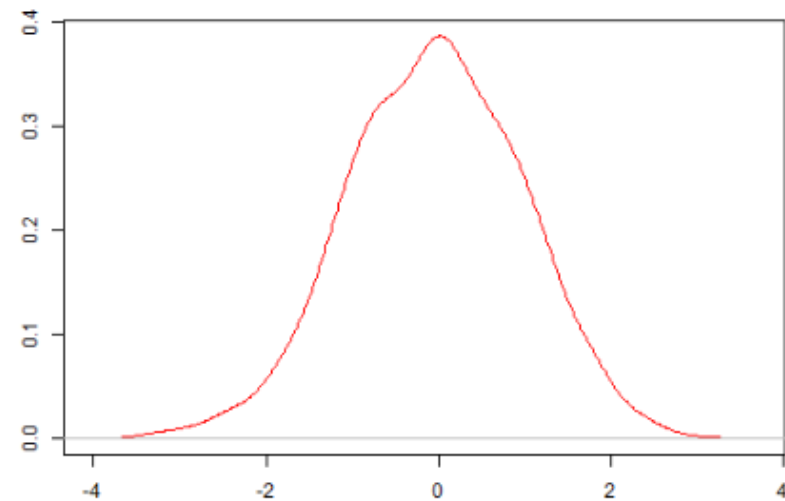


Gráfico 2



Inputs: selectInput ()

Color del gráfico 1

Azul Verde

Color del gráfico 2

Verde

```
[1] "3" "2"
```

```
[1] "green"
```

Gráfico 1

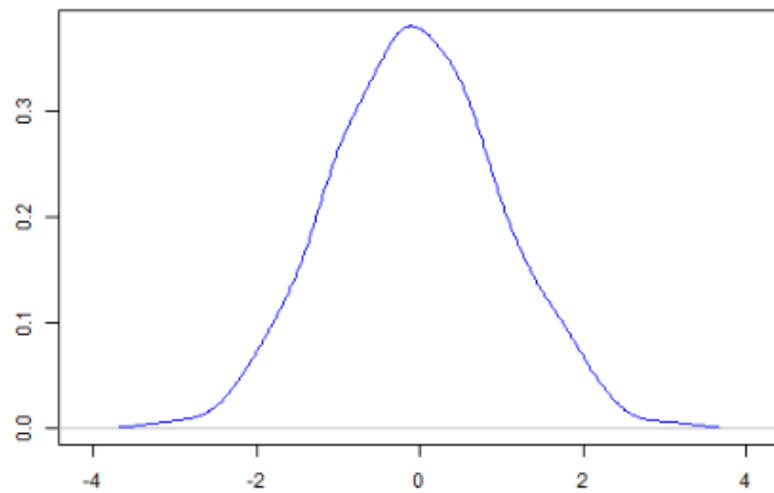
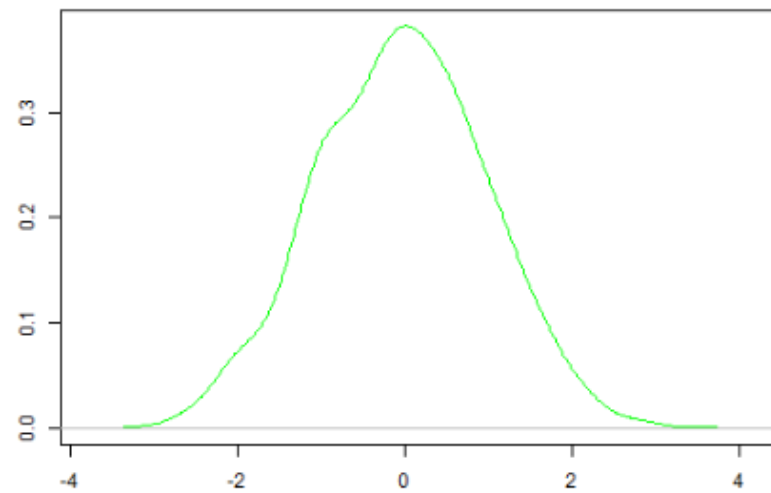
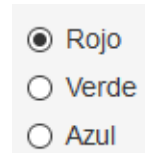


Gráfico 2



Inputs: radioButtons



☒ Rojo
☐ Verde
☐ Azul

- Permite seleccionar **una** de las opciones definidas en el control.
- Si se quiere dar la posibilidad de "no marcar" ninguna, la mejor forma es incluir una opción

Inputs: radioButtons ()

```
radioButtons(inputId, label, choices, selected = NULL, inline = FALSE,  
             width = NULL)
```

Arguments

choices: List of values to select from (if elements of the list are named then that name rather than the value is displayed to the user)

selected: The initially selected value (if not specified then defaults to the first value)

inline: If TRUE, render the choices inline (i.e. horizontally)

Las opciones `radioButtons` y `checkboxButtons` tienen el mismo comportamiento que en el control `selectButtons`. La opción `vertical` permite ordenar las opciones de forma vertical (por defecto) o de forma horizontal.

Inputs: radioButtons ()

```
inputPanel(  
  radioButtons("radio1", label = "Selecciona una opción",  
    choices = list("Rojo" = 1, "Verde" = 2, "Azul" = 3),  
    selected = 1),  
  radioButtons("radio2", label = "Selecciona una opción",  
    choices = list("Rojo" = "red", "Verde" = "green", "Azul" = "blue"),  
    selected = "red")  
)  
renderPlot({  
  par(mfrow=c(1,2))  
  plot(density(rnorm(1000,0,1)),col=as.numeric(input$radio1)+1,xlab="",ylab="",main="")  
  plot(density(rnorm(1000,0,1)),col=input$radio2,xlab="",ylab="",main="")  
})
```

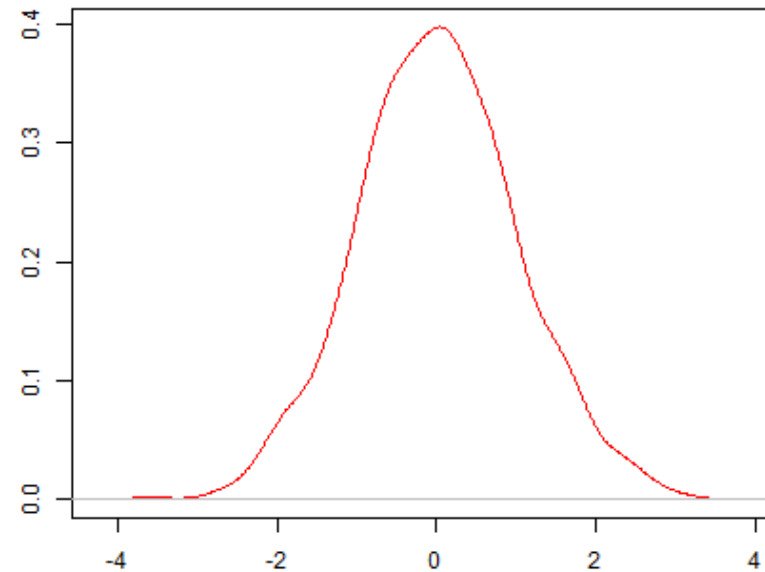
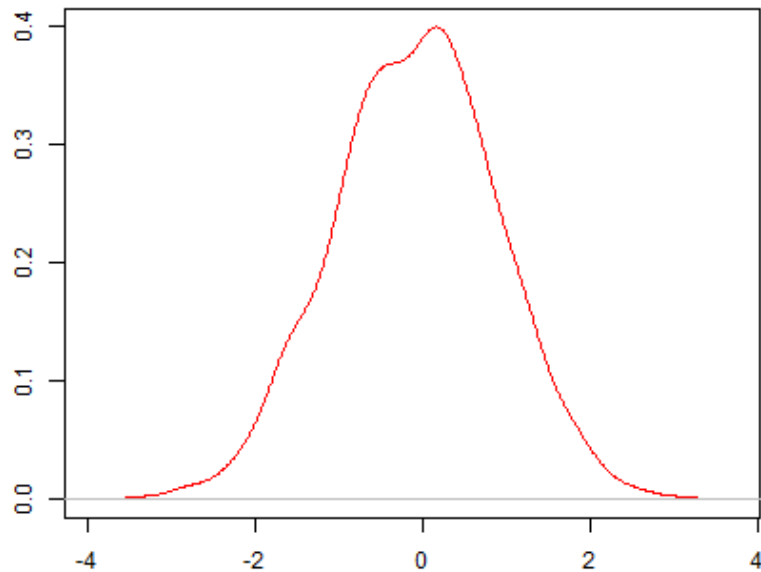
Inputs: radioButtons ()

Selecciona una opción

- ☒ Rojo
- ☐ Verde
- ☐ Azul

Selecciona una opción

- ☒ Rojo
- ☐ Verde
- ☐ Azul



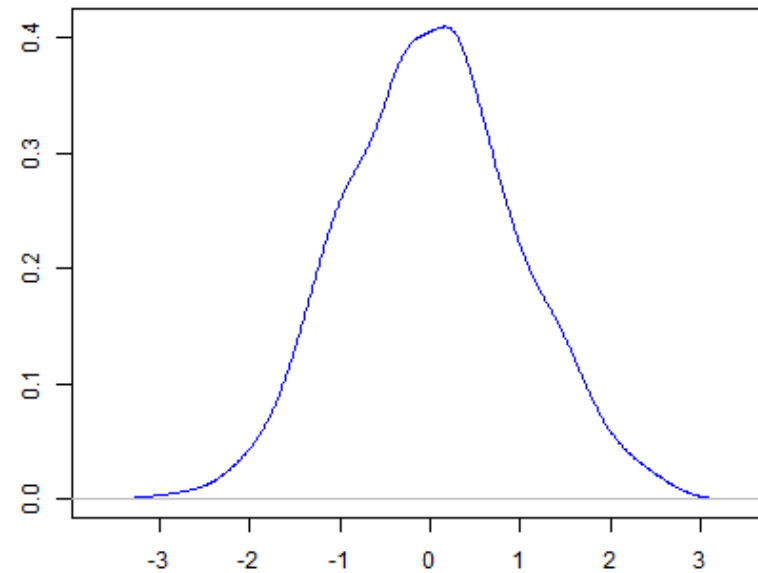
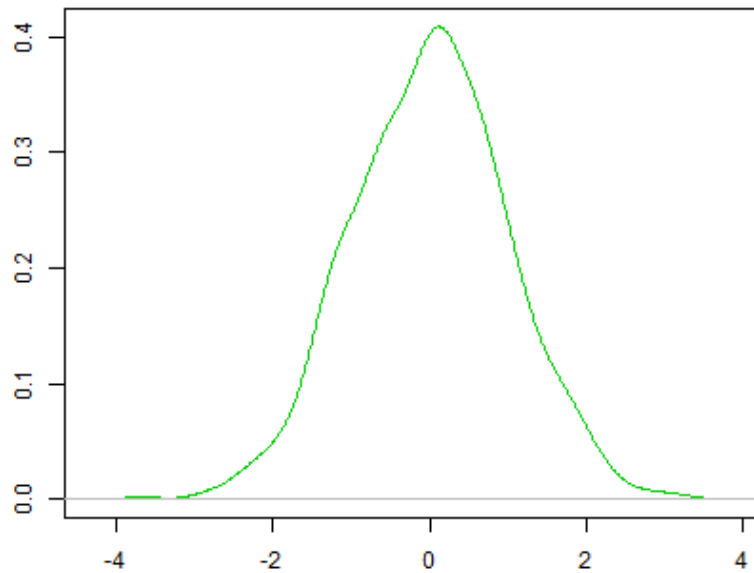
Inputs: radioButtons ()

Selecciona una opción

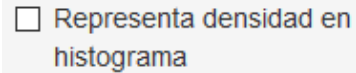
- ☐ Rojo
☒ Verde
☐ Azul

Selecciona una opción

- ☐ Rojo
☐ Verde
☒ Azul



Inputs: checkboxInput



☐ Representa densidad en histograma

- Permite marcar o no una opción propuesta.
- Toma el valor FALSE si la opción no ha sido marcada y el valor TRUE si la opción sí ha sido marcada. Por defecto podemos incluir si en un inicio queremos que esté marcada o no (parámetro value).

Arguments

```
checkboxInput(inputId, label, value = FALSE, width = NULL)
```

Inputs: checkboxInput ()

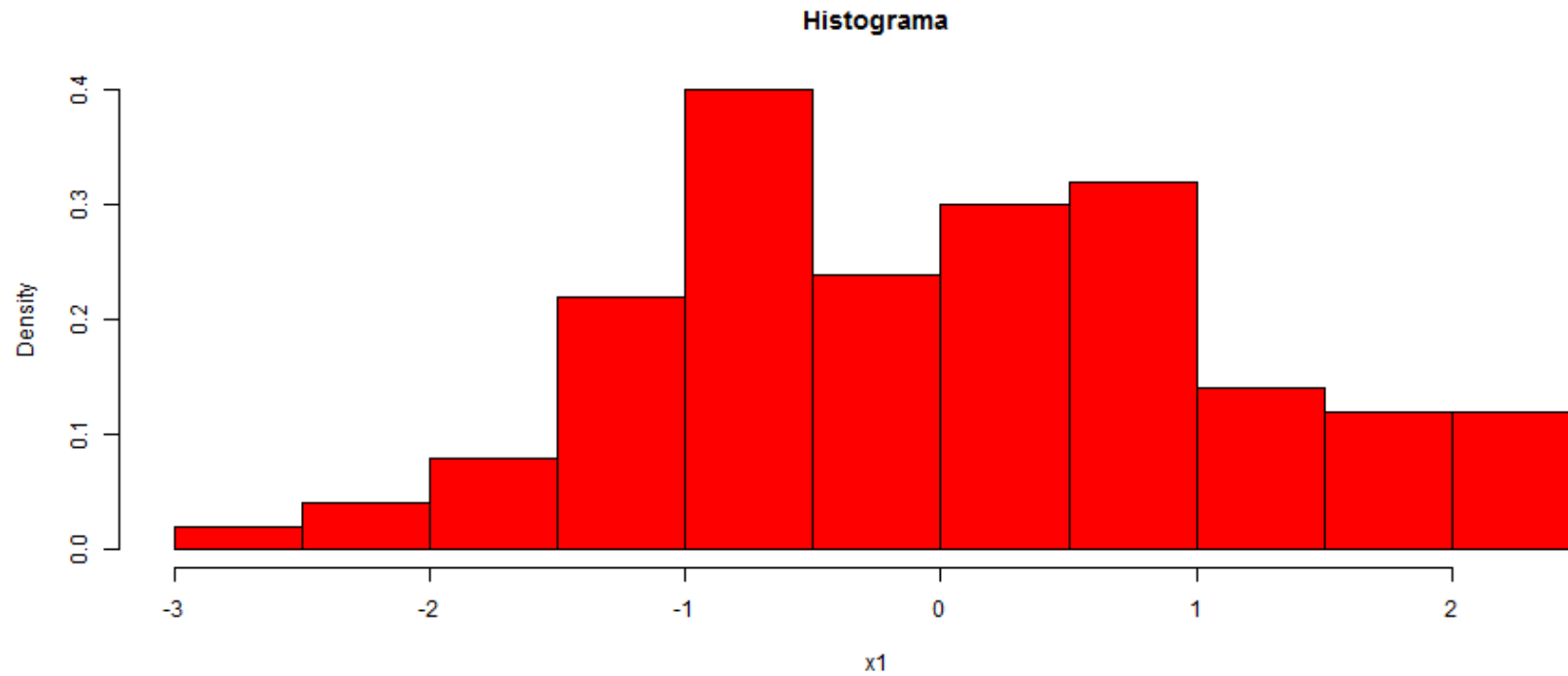
```
x1<-rnorm(100)

inputPanel(
  checkboxInput("CB1","Representa densidad en histograma")
)

renderPlot({
  hist(x1,col="red",probability=TRUE,main="Histograma")
  if (input$CB1){
    lines(density(x1),col="blue",lwd=2)
  }
})
```

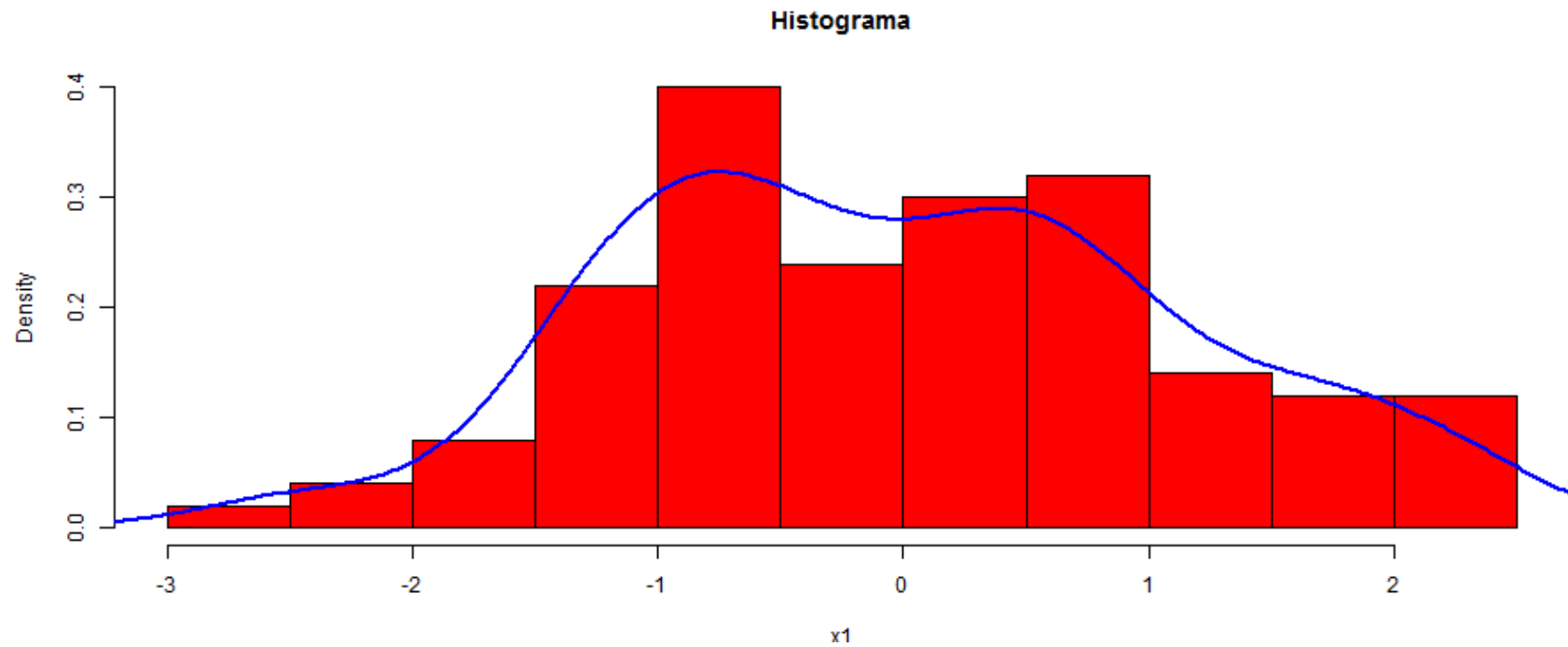
Inputs: checkboxInput ()

☐ Representa densidad en histograma

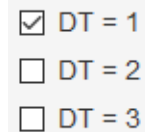


Inputs: checkboxInput ()

☒ Representa densidad en histograma



Inputs: checkboxGroupInput



☒ DT = 1
☐ DT = 2
☐ DT = 3

- Permite marcar varias de las opciones propuestas.
- Su valor es un vector con tantos elementos como opciones seleccionadas. Cada uno de esos elementos toma el valor que en la configuración se haya definido para cada opción.
- En caso de no seleccionar ninguna opción toma el valor NULL.

Inputs: checkboxGroupInput ()

```
checkboxGroupInput(inputId, label, choices, selected = NULL, inline = FALSE,  
width = NULL)
```

Arguments

choices: List of values to show checkboxes for. If elements of the list are named then that name rather than the value is displayed to the user

selected: The values that should be initially selected, if any.

inline: If TRUE, render the choices inline (i.e. horizontally)

Los parámetros choices, selected e inline funcionan de forma similar que en el control selectInput.

Inputs: checkboxGroupInput ()

```
x21<-rnorm(10000,0,1)
x22<-rnorm(10000,0,2)
x23<-rnorm(10000,0,3)
x2<-data.frame(x21,x22,x23)
colnames(x2)<-c("DV1","DV2","DV3")

inputPanel(
checkboxGroupInput("CBvariables", "Variables Normales:",
                c("DT = 1" = "DV1","DT = 2" = "DV2","DT = 3" = "DV3"),
                selected="DV1")
)

renderPrint(
  input$CBvariables
)

renderPlot({
  if(is.null(input$CBvariables)) return(NULL)
  plot(1,1,type="n",xlim=c(-10,10),ylim=c(0,0.4))
  for (i in 1:length(input$CBvariables)){
    lines(density(x2[,input$CBvariables[i]]),type="l",col=i)
  }
})
```

Inputs: checkboxGroupInput ()

Variables Normales:

☐ DT = 1

☐ DT = 2

☒ DT = 3

NULL

Inputs: checkboxGroupInput ()

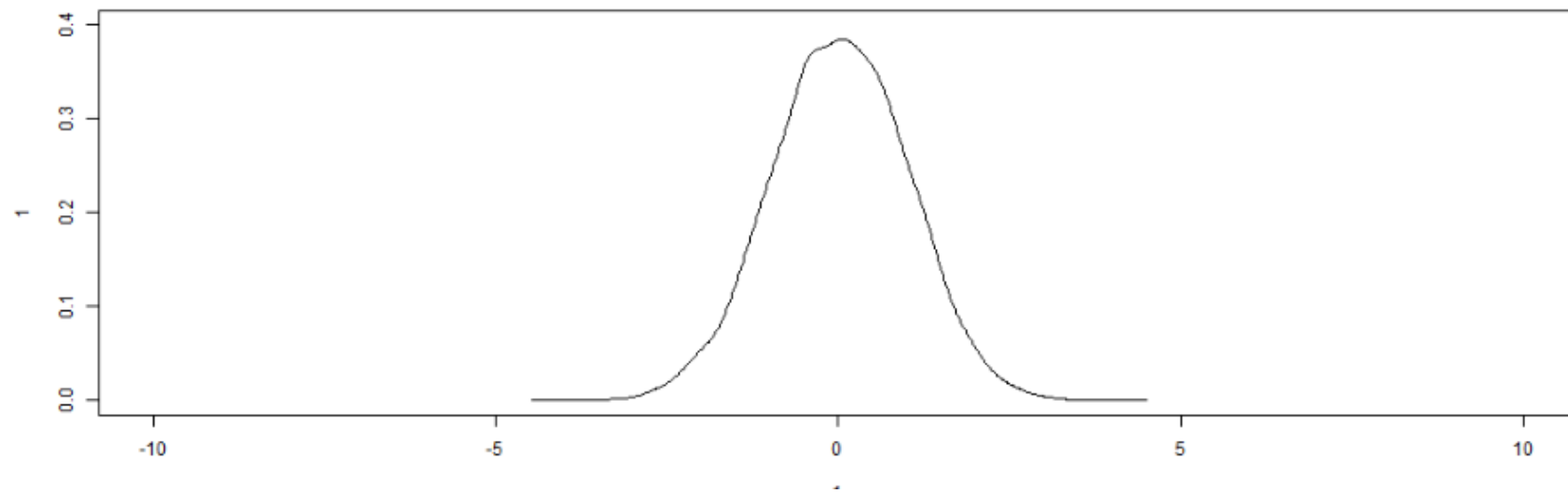
Variables Normales:

☒ DT = 1

☐ DT = 2

☐ DT = 3

[1] "DV1"



Inputs > checkboxGroupInput (Ejemplo 3)

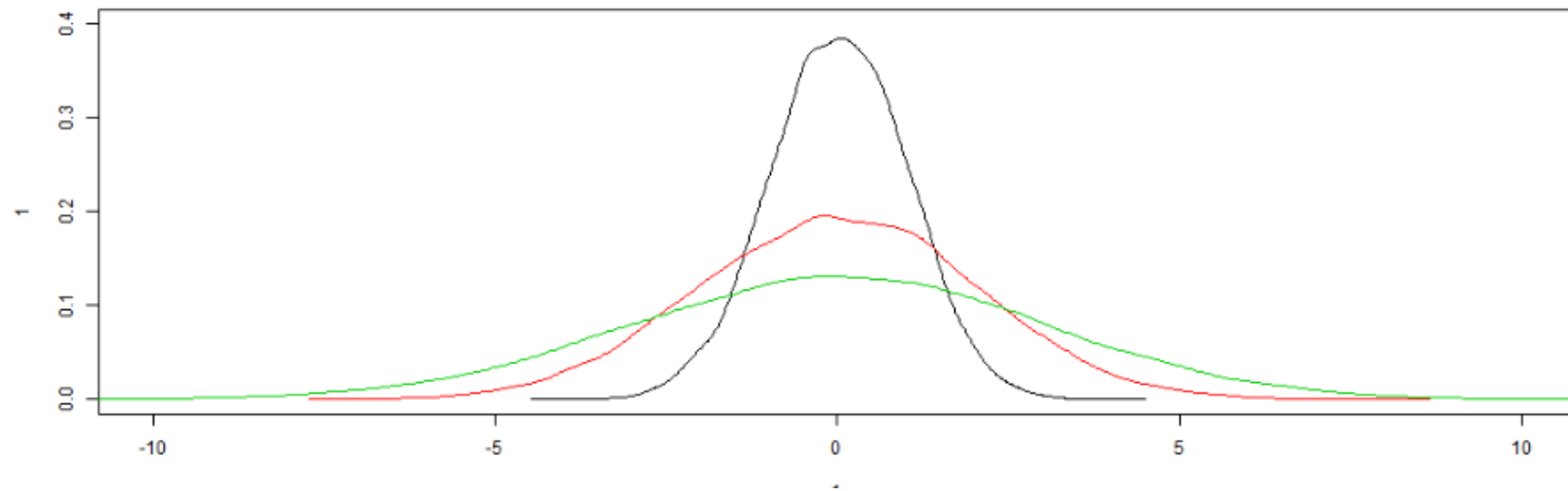
Variables Normales:

☒ DT = 1

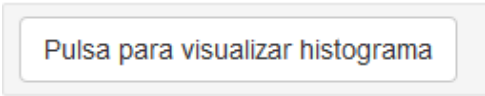
☒ DT = 2

☒ DT = 3

[1] "DV1" "DV2" "DV3"



Inputs: `actionButton`

A rectangular button with a light gray border and a white background. The text "Pulsa para visualizar histograma" is centered inside the button in a dark gray font.

- Ejecuta una acción tras ser pulsado por el usuario.
- Toma el valor por defecto, y va incrementando su valor en una unidad cada vez que es pulsado.

Inputs: `actionButton ()`

```
actionButton(inputId, label, icon = NULL, width = NULL, ...)
```

Arguments

icon: An optional icon to appear on the button.

Permite añadir un icono al botón.

Inputs: `actionButton ()`

```
inputPanel(  
  actionButton("Ejecuta","Pulsa para visualizar histograma")  
)  
  
renderPrint(  
  input$Ejecuta  
)  
renderPlot({  
  if (input$Ejecuta){  
    hist(rnorm(100),col="red",main="Histograma")  
  }  
})
```

Inputs: `actionButton ()`

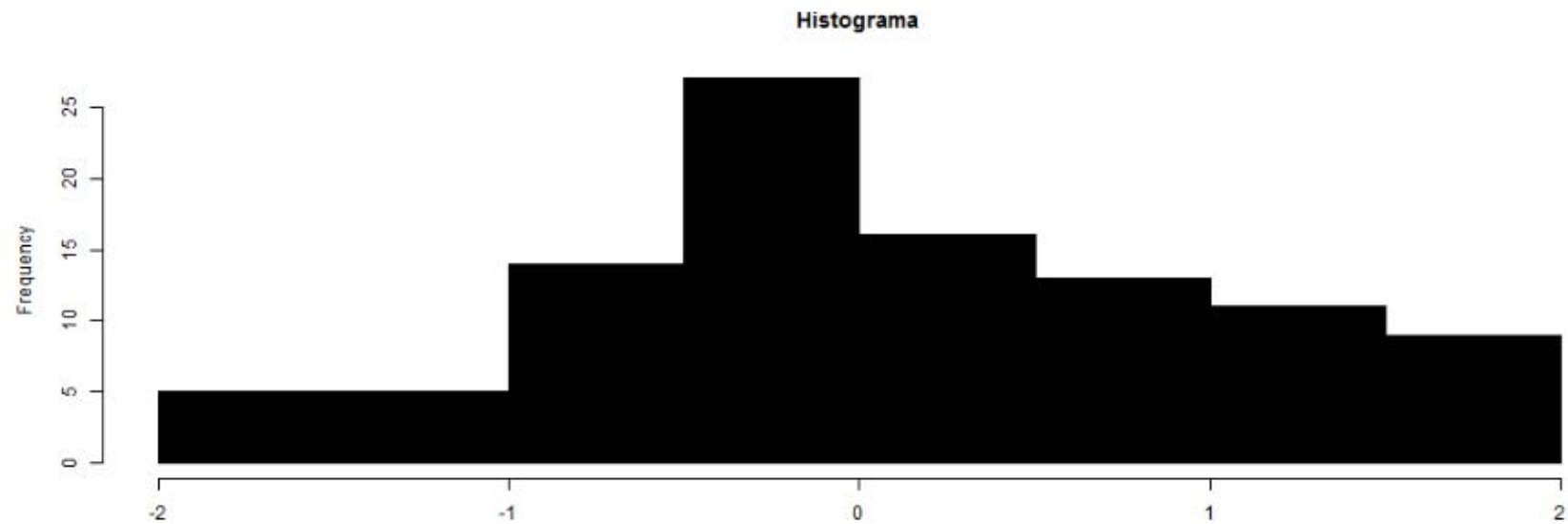
Pulsa para visualizar histograma

```
[1] 0  
attr(,"class")  
[1] "integer"           "shinyActionButtonValue"
```

Inputs: `actionButton ()`

Pulsa para visualizar histograma

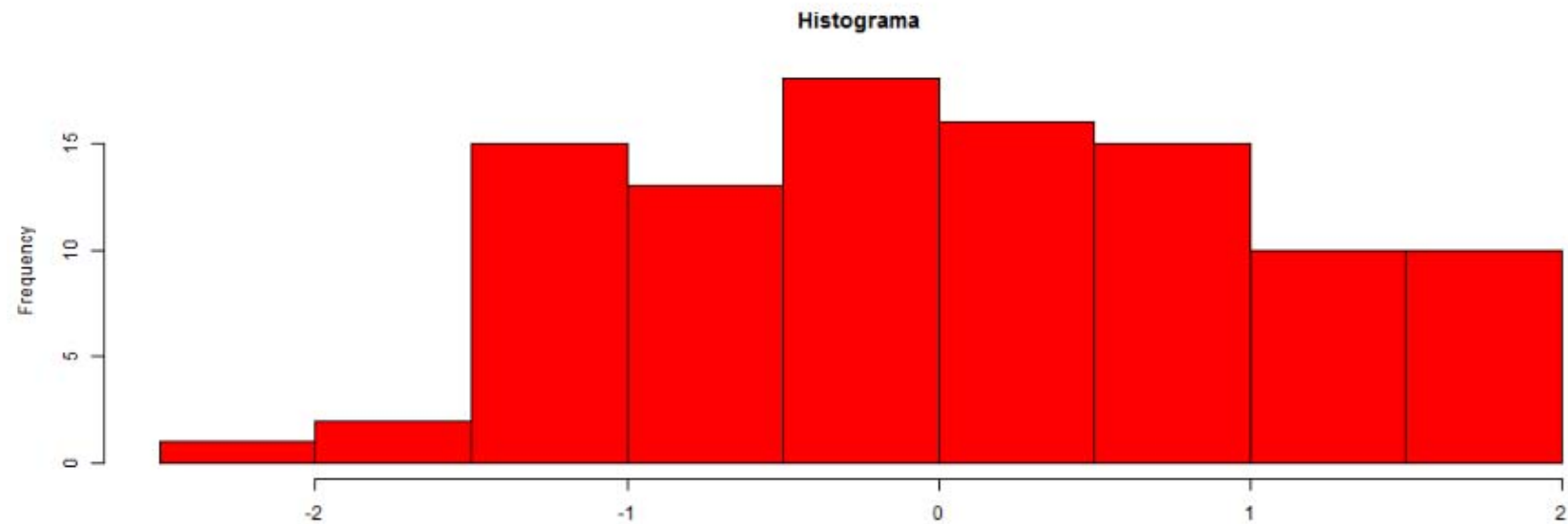
```
[1] 1  
attr(,"class")  
[1] "integer" "shinyActionButtonValue"
```



Inputs > actionButton (Ejemplo 4)

Pulsa para visualizar histograma

```
[1] 2  
attr(,"class")  
[1] "integer" "shinyActionButtonValue"
```



Inputs: fileInput ()

```
fileInput(inputId, label, multiple = FALSE, accept = NULL, width = NULL)
```

Arguments

multiple: Whether the user should be allowed to select and upload multiple files at once. Does not work on older browsers, including Internet Explorer 9 and earlier.

Mediante esta opción indicamos si permitimos cargar más de un fichero o no.

(En el parámetro `accept` se puede indicar el tipo de fichero que espera, aunque el explorador de RStudio, al menos, ignora este parámetro y permite seleccionar cualquier tipo de fichero)

Inputs: fileInput ()

Details

El programa avisa cuando la carga del fichero, o ficheros, se ha completado. La variable asociada a este es un en el que podemos encontrar una fila por cada fichero cargado y varias columnas cuyos nombres y contenidos son los siguientes:

name: Nombre del fichero

size: Tamaño del fichero (bytes)

type: El tipo de fichero (que devuelve el explorador) o vacío si no es conocido.

datapath: La ruta de acceso a un archivo temporal que contiene los datos cargados (este archivo temporal desaparecerá si el usuario realiza otra operación de carga).

Inputs: fileInput ()

```
inputPanel(  
  fileInput("fichero", label ="Selecciona un fichero csv",multiple = FALSE)  
)  
renderPrint(  
  input$fichero  
)  
renderPrint(  
  names(input$fichero)  
)  
renderPrint({  
  if(is.null(input$fichero)) return(NULL)  
  head(read.csv(input$fichero$datapath,sep=";"))  
})
```

Inputs: fileInput ()

Seleziona un fichero csv

Choose File No file selected

NULL

NULL

NULL

Inputs: fileInput ()

Selecciona un fichero csv

Choose File ...sNacido2015.csv

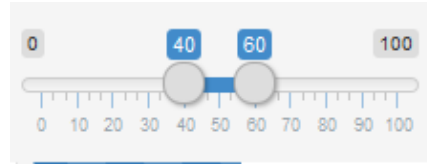
Upload complete

| | name | size | type | | datapath |
|---|-----------------------------------------------------------------------------------|------|----------|--|----------|
| 1 | NombresNacido2015.csv | 2556 | text/csv | | |
| 1 | C:\\Users\\Usuario\\AppData\\Local\\Temp\\RtmpIbn8jG\\c0cbf3da5f80708f9bf099c9\\0 | | | | |

| [1] | "name" | "size" | "type" | "datapath" |
|-----|--------|--------|--------|------------|
|-----|--------|--------|--------|------------|

| | NombreChico | TotalNombreChico | NombreChica | TotalNombreChica |
|---|-------------|------------------|-------------|------------------|
| 1 | HUGO | 5.162 | LUCIA | 5.229 |
| 2 | DANIEL | 4.578 | MARIA | 4.516 |
| 3 | PABLO | 4.202 | MARTINA | 4.447 |
| 4 | MARTIN | 4.078 | PAULA | 3.854 |
| 5 | ALEJANDRO | 3.994 | SOFIA | 3.525 |
| 6 | ADRIAN | 3.341 | DANIELA | 3.392 |

Inputs: sliderInput



- Permite seleccionar un valor en un rango propuesto o dos valores que definen un rango (también en un rango inicial propuesto).
- Toma un valor (el seleccionado) o un vector con dos valores en el caso de poder seleccionar dos valores.
- El hecho de poder seleccionar uno o dos valores depende de cómo se defina el control.

Inputs: sliderInput ()

```
sliderInput(inputId, label, min, max, value, step = NULL, round = FALSE,  
  format = NULL, locale = NULL, ticks = TRUE, animate = FALSE,  
  width = NULL, sep = ",", pre = NULL, post = NULL, timeFormat = NULL,  
  timezone = NULL, dragRange = TRUE)
```

Arguments

min: The minimum value (inclusive) that can be selected.

max: The maximum value (inclusive) that can be selected.

Valores mínimo y máximo del selector.

Inputs: sliderInput ()

```
sliderInput(inputId, label, min, max, value, step = NULL, round = FALSE,  
  format = NULL, locale = NULL, ticks = TRUE, animate = FALSE,  
  width = NULL, sep = ",", pre = NULL, post = NULL, timeFormat = NULL,  
  timezone = NULL, dragRange = TRUE)
```

value: The initial value of the slider. A numeric vector of length one will create a regular slider; a numeric vector of length two will create a double-ended range slider. A warning will be issued if the value doesn't fit between min and max.

Este valor inicial determina el tipo de selector. Si se da un único valor (que debe estar dentro del rango establecido) el selector permitirá elegir un solo valor, si se proporciona un vector con dos valores (también dentro del rango establecido), el selector permitirá elegir un par de valores que definirán el rango seleccionado.

Inputs: sliderInput ()

```
sliderInput(inputId, label, min, max, value, step = NULL, round = FALSE,  
  format = NULL, locale = NULL, ticks = TRUE, animate = FALSE,  
  width = NULL, sep = ",", pre = NULL, post = NULL, timeFormat = NULL,  
  timezone = NULL, dragRange = TRUE)
```

Arguments

step: Specifies the interval between each selectable value on the slider (if NULL, a heuristic is used to determine the step size). If the values are dates, step is in days; if the values are times (POSIXt), step is in seconds.

En este parámetro indicaremos el entre dos valores consecutivos de la barra selectora (saltos de uno en uno, dos en dos,...). Determinará qué valores se podrán elegir y aquellos que no podrán ser elegidos.

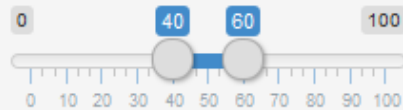
Inputs: sliderInput ()

```
inputPanel(  
  sliderInput("slider1", label = h3("Selecciono rango mínimo y máximo"),  
    min = 0, max = 100,  
    value = c(40,60))  
)  
renderPrint({  
  input$slider1  
})
```

```
inputPanel(  
  sliderInput("slider2", label = h3("Selecciono rango máximo"),  
    min = 0, max = 100,  
    value = 50)  
)  
renderPrint({  
  input$slider2  
})
```

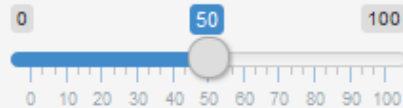
Inputs: sliderInput ()

Selecciono rango
mínimo y máximo



```
[1] 40 60
```

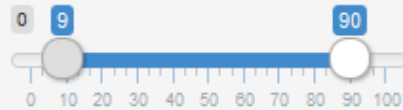
Selecciono rango
máximo



```
[1] 50
```

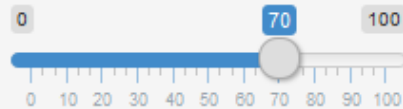
Inputs: sliderInput ()

Selecciono rango
mínimo y máximo



```
[1] 9 90
```

Selecciono rango
máximo



```
[1] 70
```

Widgets: Outputs

Widgets: Outputs

Rendered output

To add reactive output to your document, call one of the `render*` functions below in an R code chunk.

render function creates

| | |
|--------------------------|-------------------------------------------------|
| <code>renderImage</code> | images (saved as a link to a source file) |
| <code>renderPlot</code> | plots |
| <code>renderPrint</code> | any printed output |
| <code>renderTable</code> | data frame, matrix, other table like structures |
| <code>renderText</code> | character strings |
| <code>renderUI</code> | a Shiny tag object or HTML |

De la misma forma que disponemos de unos objetos que permitirán al usuario seleccionar algunas opciones que determinarán la forma o contenido del documento interactivo (inputs), disponemos de diferentes opciones para presentar resultados de texto, numéricos o gráficos (outputs).

Outputs: renderPrint ()

Útil para mostrar cualquier salida de resultados (de un banco de datos o resultado de un análisis estadístico)

```
renderPrint(expr, env = parent.frame(), quoted = FALSE, func = NULL,  
            width = getOption("width"))
```

Arguments

expr: An expression that may print output and/or return a printable R object.

Esta salida, como hemos visto durante la introducción de los tipos de sirve para mostrar cualquier resultado numérico, de texto que pueda ser impreso.

Outputs: renderPrint ()

```
data("DNase")

inputPanel(
  selectInput(inputId = "seleccionaFun", label="Selecciona función",
    choices = c("Covarianza"=1,"Correlación"=2),selected = 1)
)

renderPrint({
  if(input$seleccionaFun==1){cov(DNase[,2:3])}
  else{cor(DNase[,2:3])}
})
```


Outputs: renderPrint ()

| Selecciona función | | |
|--------------------|-----------|-----------|
| Covarianza ▼ | | |
| | conc | density |
| conc | 16.482506 | 2.2510272 |
| density | 2.251027 | 0.3547067 |

Outputs: `renderText ()`

```
renderText(expr, env = parent.frame(), quoted = FALSE, func = NULL)
```

Arguments

expr: An expression that returns an R object that can be used as an argument to `cat`.

En este tipo de salida podemos mostrar cualquier texto que pueda ser mostrado en R mediante la función `cat`. Puede incluir varias líneas incorporando el carácter `\n`.

Outputs: renderText ()

```
data("DNase")  
  
renderText({  
  names(input) #imprimimos contenido objeto input  
})
```

renderText

CB1 CBvariables ColorSel1 ColorSel2 Ejecuta etiqx etiqy Fecha FechaDesdeHasta fichero num radio1 radio2 seleccionaFun slider1 slider2 titulo

Outputs: renderPlot ()

```
renderPlot(expr, width = "auto", height = "auto", res = 72, ...,  
  env = parent.frame(), quoted = FALSE, func = NULL)
```

Arguments

expr: An expression that generates a plot.

width, height: The width/height of the rendered plot, in pixels; or 'auto' to use the offsetWidth/offsetHeight of the HTML element that is bound to this plot. ...

res: Resolution of resulting plot, in pixels per inch. This value is passed to png. Note that this affects the resolution of PNG rendering in R; it won't change the actual ppi of the browser.

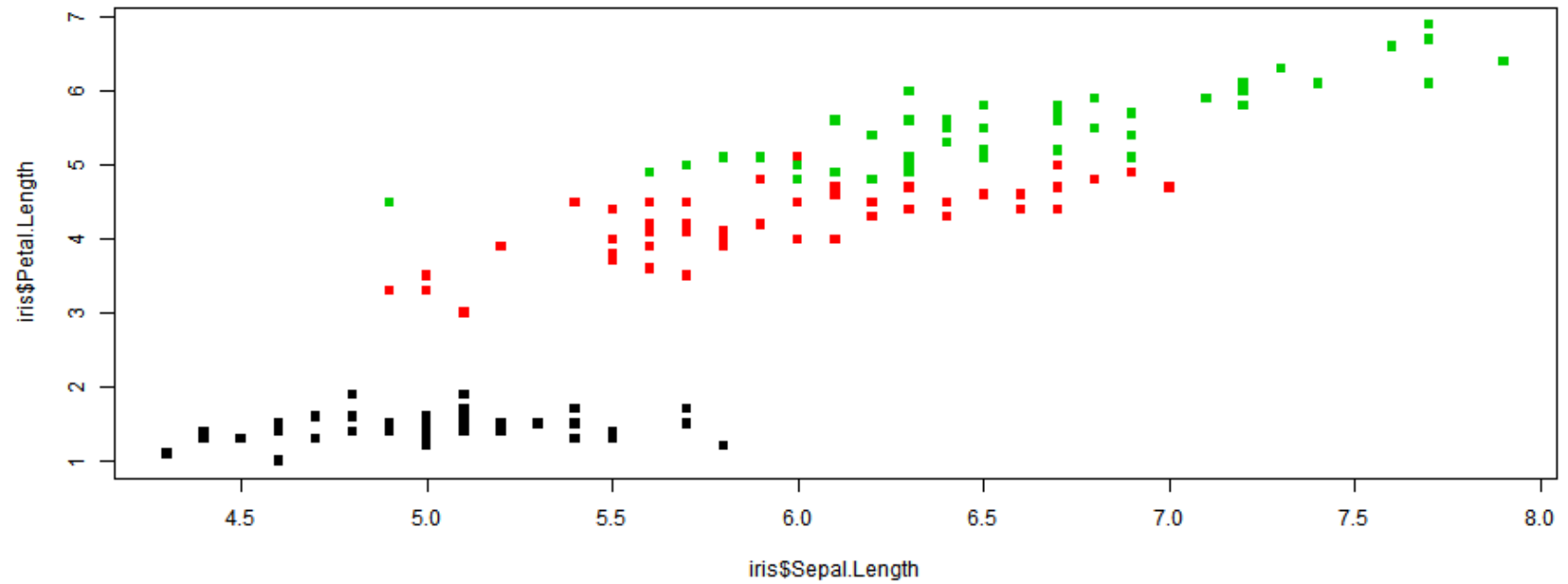
Este tipo de salida permite alojar cualquier tipo de gráfico.

Outputs: renderPlot ()

```
data("iris")

renderPlot({
  plot(iris$Sepal.Length,iris$Petal.Length,pch=15,col=iris$Species)
})
```

Outputs: renderPlot ()



Outputs: renderTable ()

```
renderTable(expr, ..., env = parent.frame(), quoted = FALSE, func = NULL)
```

Arguments

expr: An expression that returns an R object that can be used with `xtable`.

Cualquier objeto `matrix`, `data.frame`,...o la salida de la función `xtable`, que permite transformar en objeto tabla de forma sencilla por ejemplo salidas de resultados de la aplicación de modelos lineales,...

```
xtable(x, caption = NULL, label = NULL, align = NULL, digits = NULL,  
       display = NULL, auto = FALSE, ...)
```

(Explorar también [xtable](#), que introduce una tabla con funcionalidades de orden, búsqueda, selección,...Se verá en sesión de **Shiny**)

Outputs: renderTable ()

```
data("iris")  
  
renderTable({  
  Iris  
})
```


Outputs: renderTable ()

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|----|--------------|-------------|--------------|-------------|---------|
| 1 | 5.10 | 3.50 | 1.40 | 0.20 | setosa |
| 2 | 4.90 | 3.00 | 1.40 | 0.20 | setosa |
| 3 | 4.70 | 3.20 | 1.30 | 0.20 | setosa |
| 4 | 4.60 | 3.10 | 1.50 | 0.20 | setosa |
| 5 | 5.00 | 3.60 | 1.40 | 0.20 | setosa |
| 6 | 5.40 | 3.90 | 1.70 | 0.40 | setosa |
| 7 | 4.60 | 3.40 | 1.40 | 0.30 | setosa |
| 8 | 5.00 | 3.40 | 1.50 | 0.20 | setosa |
| 9 | 4.40 | 2.90 | 1.40 | 0.20 | setosa |
| 10 | 4.90 | 3.10 | 1.50 | 0.10 | setosa |
| 11 | 5.40 | 3.70 | 1.50 | 0.20 | setosa |
| 12 | 4.80 | 3.40 | 1.60 | 0.20 | setosa |
| 13 | 4.80 | 3.00 | 1.40 | 0.10 | setosa |

Outputs > renderImage (Sintaxis)

```
renderImage(expr, env = parent.frame(), quoted = FALSE, deleteFile = TRUE)
```

Arguments

expr: An expression that returns a list.

deleteFile: Should the file in `func()$src` be deleted after it is sent to the client browser? Generally speaking, if the image is a temp file generated within `func`, then this should be `TRUE`; if the image is not a temp file, this should be `FALSE`.

Outputs: `renderUI ()`

```
renderUI(expr, env = parent.frame(), quoted = FALSE, func = NULL)
```

Arguments

expr: An expression that returns a Shiny tag object, HTML, or a list of such objects.

Este tipo de output permite alojar un objeto de tipo aplicación shiny o a su vez otro documento interactivo.

Para terminar...

- En esta sesión hemos introducido los elementos básicos de **shiny** que pueden ser incorporados en un documento **RMarkdown** de forma muy sencilla y que nos permiten convertir en **interactivos** nuestros documentos.
- En la sesión de **shiny** veremos cómo construir estructuras para alojar y organizar los input y los output, así como tipos de output más sofisticados e interactivos.
- Estamos ante unas herramientas , que día a día incorporan nuevos elementos y nuevas posibilidades de desarrollo.

Recursos útiles:

- [Interactive Documents \(http://rmarkdown.rstudio.com/lesson-14.html\)](http://rmarkdown.rstudio.com/lesson-14.html)
- [El R Markdown Cheatsheet \(https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-spanish.pdf\)](https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-spanish.pdf).
- [Input Widgets \(https://shiny.rstudio.com/tutorial/lesson3/\)](https://shiny.rstudio.com/tutorial/lesson3/)
- [Widgets gallery \(https://shiny.rstudio.com/gallery/widget-gallery.html\)](https://shiny.rstudio.com/gallery/widget-gallery.html)

Referencias

Cirillo, Andrea. 2016. . Packt Publishing Ltd.

Hillebrand, Julian, and Maximilian H Nierhoff. 2015.
. Packt Publishing Ltd.

Resnizky, Hernán G. 2015. . Packt Publishing Ltd.

