

R Markdown

Escribiendo informes reproducibles

Paqui Corpas Burgos - Área de Desigualdades en Salud, FISABIO

6 de Junio, 2017

Contenidos

1. [Introducción](#)
2. [Creación de un documento R Markdown desde RStudio](#)
3. [Componentes de un archivo R Markdown](#)
4. [Formateando el texto con markdown](#)
5. [Trozos de código R](#)
6. [Opciones globales](#)
7. [Almacenamiento en caché](#)
8. [Tablas](#)
9. [Figuras](#)
10. [Formatos de salida](#)
11. [Informes parametrizados](#)
12. [Bibliografía y citas](#)

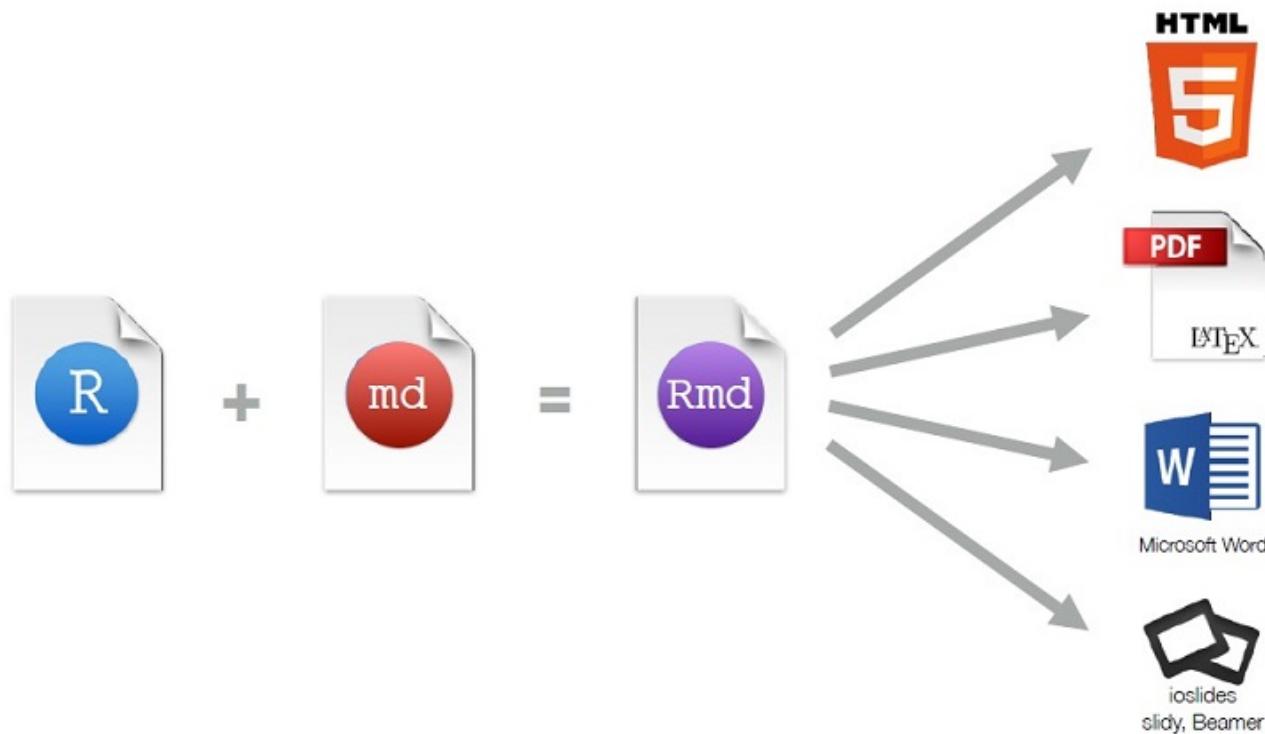
1. Introducción: ¿Qué es R Markdown?

Los analistas de datos a menudo escribimos informes presentando los resultados obtenidos tras el análisis estadístico de unos datos. Es conveniente que los **resultados** presentados sean completamente **reproducibles**.

R Markdown (<http://rmarkdown.rstudio.com/>) permite combinar texto, código y los resultados del código en un sólo documento. Al procesar el documento, el código se evalúa y los resultados obtenidos se presentan junto al texto en el informe.

El texto se escribe en lenguaje Markdown (<http://daringfireball.net>), una forma sencilla de escribir texto simple (secciones, párrafos, listas, enlaces, imágenes,...) que se convierte fácilmente a otros formatos (HTML, PDF, Word, ...).

R Markdown



Fuente (<https://github.com/rstudio/webinars/blob/master/12-Getting-started-with-R-Markdown/12-Getting-started-with-R-Markdown.pdf>)

Introducción: Utilidades de R Markdown

- **Procesador de textos:** escribir documentos que contengan fórmulas matemáticas, tablas y figuras.
- Presentar, compartir y publicar **resultados en varios formatos** (pdf, word, html, ...).
- Generar informes **automáticamente**.
- Organizar y compartir **código y resultados en un mismo documento** fácil de escribir y leer.
 - Tablas y figuras sincronizadas con la última versión de los datos y el código.
 - Se reducen errores de transcripción manual de resultados.
- Herramienta muy útil para la reproducibilidad (<https://en.wikipedia.org/wiki/Reproducibility>) y para todo usuario de R.

☞ **Objetivo:** Aprender a elaborar informes reproducibles utilizando R Markdown.

Introducción: ¿Cómo funciona R Markdown?

La creación de documentos a partir de R Markdown comienza con un archivo **.Rmd** que contiene texto y trozos de código R. El archivo **.Rmd** se envía a **knitr** (<https://yihui.name/knitr/>), que ejecuta los fragmentos de código R y crea un documento markdown (**.md**) que incluye el código y su salida. Entonces, **Pandoc** (<http://pandoc.org/>) procesa el archivo **.md** para crear el informe final en forma de página web, PDF, documento Word, presentación, etc.



Fuente (<http://rmarkdown.rstudio.com/lesson-2.html>)

El paquete **Rmarkdown** encapsula todo este proceso en una función llamada **render()**.

R Markdown



Markdown

R Markdown rocks

This is an R Markdown document.

```
```{r}
x <- rnorm(1000)
head(x)
```
```

See how the R code gets executed and a representation thereof appears in the document? `knitr` gives you control over how to represent all conceivable types of output. In case you care, then average of the `r length(x)` random normal variates we just generated is `r round(mean(x), 3)`. Those numbers are NOT hard-wired but are computed on-the-fly. As is this figure. No more copy-paste ... copy-paste ... oops forgot to copy-paste.

```
```{r}
plot(density(x))
```
```

Note that all the previously demonstrated math typesetting still works. You don't have to choose between having math cred and being web-friendly!

Inline equations, such as ... the average is computed as $\frac{1}{n} \sum_{i=1}^n x_i$. Or display equations like this:

```
$$
\begin{equation*}
|x| =
\begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x \leq 0 \end{cases}
\end{equation*}
$$
```

R Markdown rocks

This is an R Markdown document.

```
```{r}
x <- rnorm(1000)
head(x)
```
```

```
...
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973
2.4157
...  
...
```

See how the R code gets executed and a representation thereof appears in the document? `knitr` gives you control over how to represent all conceivable types of output. In case you care, then average of the 1000 random normal variates we just generated is -0.081. Those numbers are NOT hard-wired but are computed on-the-fly. As is this figure. No more copy-paste ... copy-paste ... oops forgot to copy-paste.

```
...
plot(density(x))
...  
...
```

```
![plot of chunk unnamed-chunk-2](figure/unnamed-
chunk-2.png)
```

...

Markdown → HTML

```
R Markdown rocks
```

```
=====
```

```
This is an R Markdown document.
```

```
```r
x <- rnorm(1000)
head(x)
```
```
[1] -1.3007 0.7715 0.5585 -1.2854 1.1973
2.4157
```

```

See how the R code gets executed and a representation thereof appears in the document? `knitr` gives you control over how to represent all conceivable types of output. In case you care, then average of the 1000 random normal variates we just generated is -0.081. Those numbers are NOT hard-wired but are computed on-the-fly. As is this figure. No more copy-paste ... copy-paste ... oops forgot to copy-paste.

```
```r
plot(density(x))
```

```

```
![plot of chunk unnamed-chunk-2](figure/unnamed-
chunk-2.png)
```

```
...
```

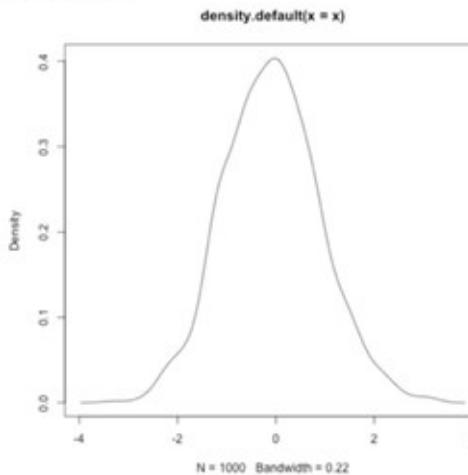
R Markdown rocks

This is an R Markdown document.

```
x <- rnorm(1000)
head(x)
```

```
## [1] -1.3007  0.7715  0.5585 -1.2854  1.1973  2.4157
```

See how the R code gets executed and a representation thereof appears in the document? knitr gives you control over how to represent all conceivable types of output. In case you care, then average of the 1000 random normal variates we just generated is -0.081. Those numbers are NOT hard-wired but are computed on-the-fly. As is this figure. No more copy-paste ... copy-paste ... oops forgot to copy-paste.



Note that all the previously demonstrated math typesetting still works. You don't have to choose between having math cred and being web-friendly!

Inline equations, such as ... the average is computed as $\frac{1}{n} \sum_{i=1}^n x_i$. Or display equations like this:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x \leq 0. \end{cases}$$

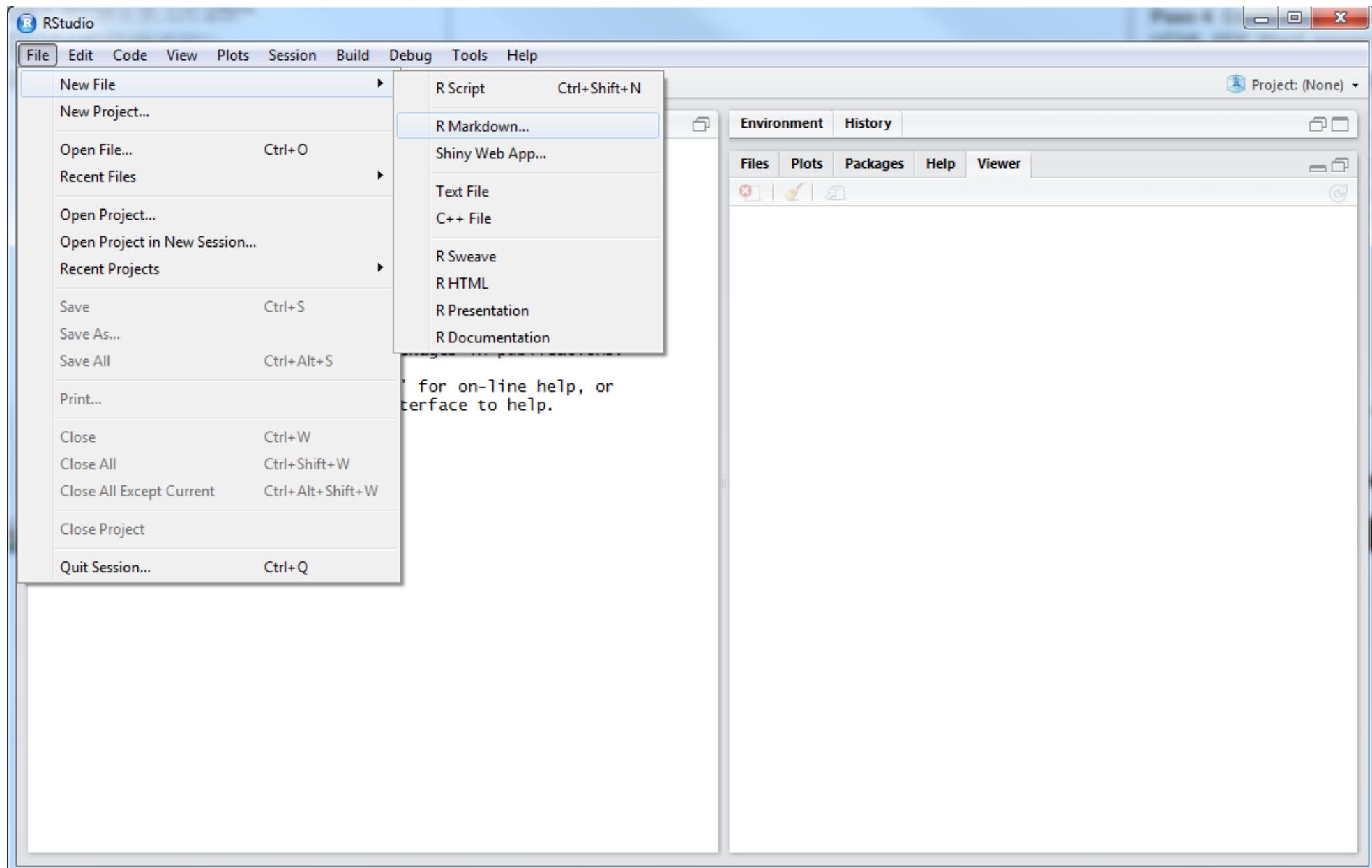
2. Creación de un documento R Markdown desde RStudio

Paso 1. Abrimos RStudio.

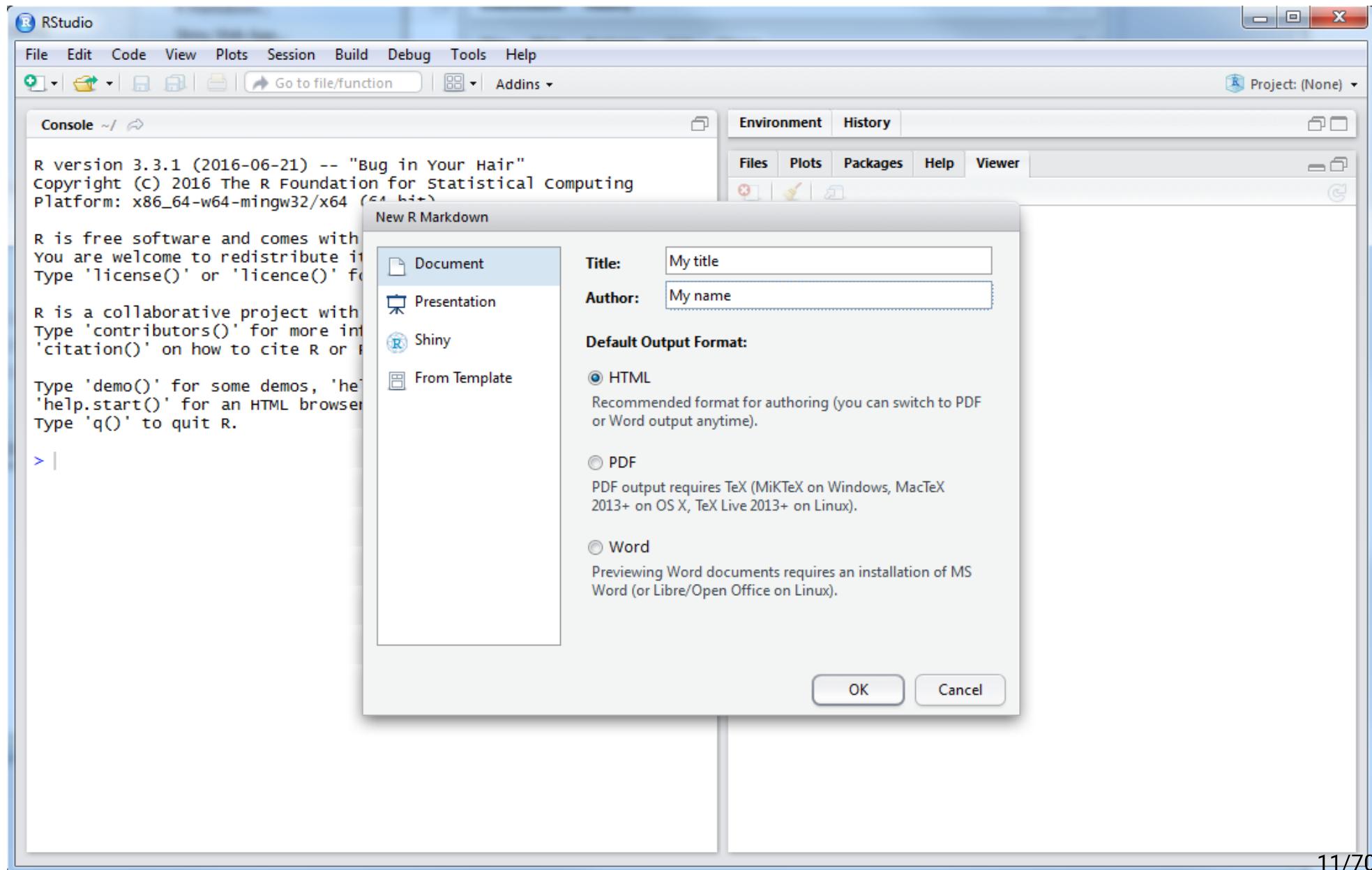
Paso 2. Instalamos el paquete R Markdown desde CRAN (en caso de no haberlo instalado previamente):

```
install.packages("rmarkdown")
```

Paso 3. Seleccionamos File ⇒ New File ⇒ R Markdown.



Paso 4. Escribimos el título, autor, elegimos el formato de nuestro documento: HTML, PDF, Word, presentación, ... y pulsamos OK.



A continuación, se genera automáticamente un documento R Markdown de ejemplo. Para obtener el informe final utilizamos la función `render("Ejemplo.Rmd")` o pulsamos el botón **knit**.

The screenshot shows the RStudio interface with the following components:

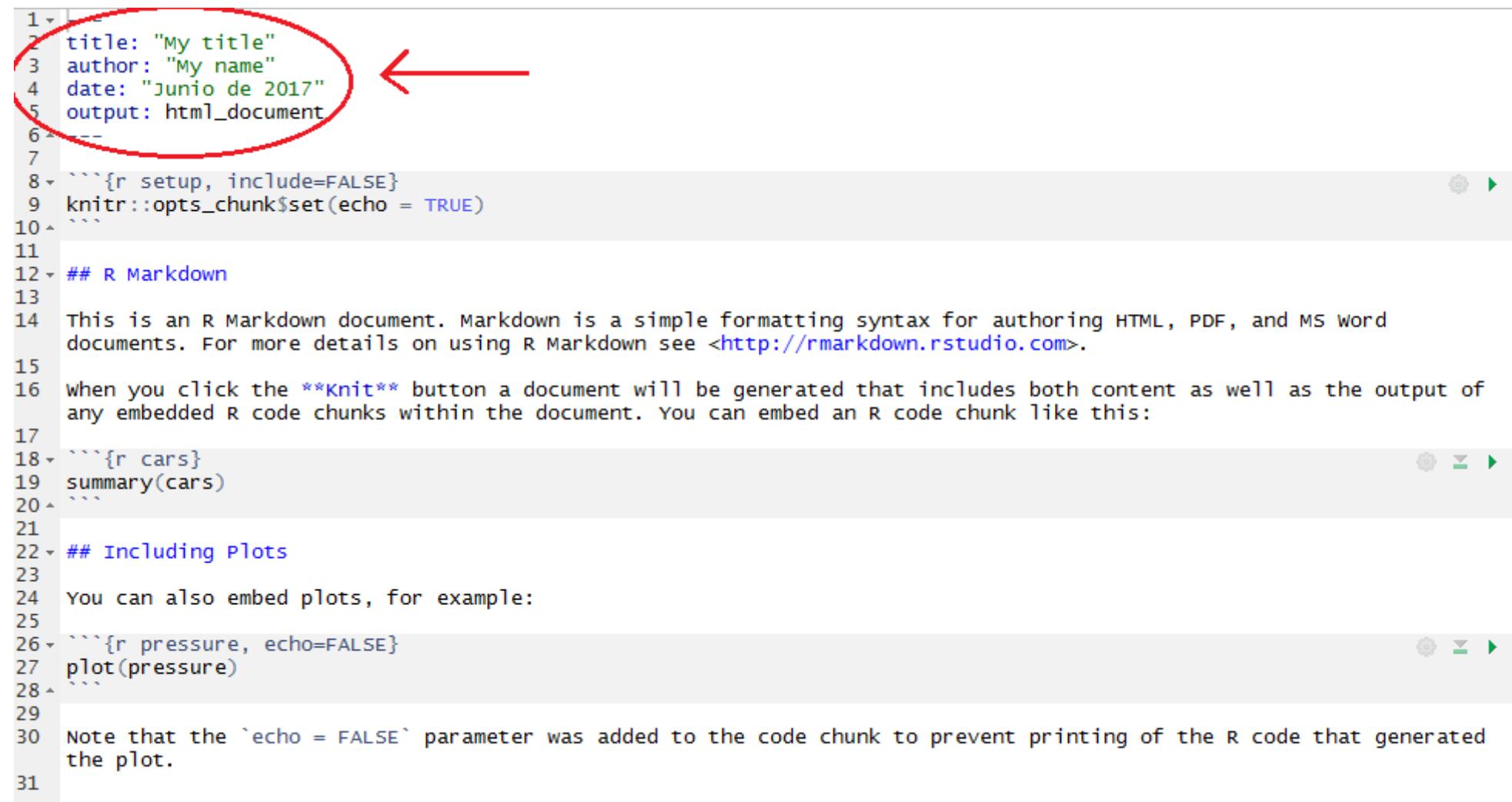
- File Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Tools, Help.
- Toolbar:** Includes icons for New, Open, Save, Print, Go to file/function, and Addins.
- Project Bar:** Project: (None).
- Code Editor:** Shows the R Markdown file "Ejemplo.Rmd". A red circle highlights the "Knit HTML" button in the toolbar above the editor, with a red arrow pointing to it from the left.
- Environment Tab:** Shows the current environment variables.
- Viewer Tab:** Shows the rendered content of the R Markdown document.
- Output Content:** The rendered document includes:
 - Section Headers:** "My title" and "My name".
 - Text:** "This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <<http://rmarkdown.rstudio.com>>."
 - Text:** "when you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
 - Code Block:** `summary(cars)`
 - Table:** A summary table for the "cars" dataset:

| | speed | dist |
|------------|--------|----------------|
| ## Min. | : 4.0 | Min. : 2.00 |
| ## 1st Qu. | : 12.0 | 1st Qu.: 26.00 |
| ## Median | : 15.0 | Median : 36.00 |
| ## Mean | : 15.4 | Mean : 42.98 |
| ## 3rd Qu. | : 19.0 | 3rd Qu.: 56.00 |
| ## Max. | : 25.0 | Max. : 120.00 |
 - Section Headers:** "## Including Plots" and "Including Plots".
- Console:** Shows the command "R Markdown".
- Page Number:** 12/70.

3. Componentes de un archivo R Markdown

Hay 3 componentes generales en un archivo R Markdown:

- Cabecera (YAML) rodeada por ---



```
1 <!--
2   title: "My title"
3   author: "My name"
4   date: "Junio de 2017"
5   output: html_document
6 ---
7
8 `r setup, include=FALSE`
9 knitr::opts_chunk$set(echo = TRUE)
10
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
15
16 when you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
17
18 `r cars`
19 summary(cars)
20
21
22 ## Including Plots
23
24 You can also embed plots, for example:
25
26 `r pressure, echo=FALSE`
27 plot(pressure)
28
29
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
31
```

► Texto en formato Markdown

```
1 ---  
2 title: "My title"  
3 author: "My name"  
4 date: "Junio de 2017"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ````  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 when you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ````  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ````  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.  
31
```



► Trozos de código R (code chunks) rodeados por :```{r} código_R```

```
1 ---  
2 title: "My title"  
3 author: "My name"  
4 date: "Junio de 2017"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ````  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 when you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ````  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ````  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.  
31
```

The image shows a screenshot of an R Markdown editor interface. Several parts of the code are highlighted with red circles and arrows:

- A red circle highlights the first code chunk from line 8 to line 10. A red arrow points from this circle to the green 'Knit' button in the top right corner.
- A red circle highlights the second code chunk from line 18 to line 20. A yellow arrow points from this circle to the green 'Knit' button in the middle right corner.
- A red circle highlights the third code chunk from line 26 to line 28. A yellow arrow points from this circle to the green 'Knit' button in the bottom right corner.

4. Formateando el texto: sintaxis en línea

- Texto en *cursiva* o cursiva ► Texto en o
- Texto en **negrita** o negrita ► Texto en **negrita** o **negrita**
- Subíndice^{~2~}, superíndice^{^2^} ► Subíndice₂, superíndice²
- ~~Tachado~~ ► ~~Tachado~~
- Enlace a [RStudio](<http://www.rstudio.com>) ► Enlace a [RStudio](http://www.rstudio.com) (<http://www.rstudio.com>)
- Imagen: :

Imagen:



- 'Texto como código' ► Texto como código
- ^[Nota a pie de página] ► ¹[_](#)

```
---
```

```
title: "Sintaxis R Markdown"
output: html_document
---
```

```
***
```

```
## Formato en línea
```

```
***
```

```
Texto en *cursiva* o _cursiva_
```

```
Texto en **negrita** o __negrita__
```

```
Subíndice~2~
```

```
Superíndice2
```

```
~~Tachado~~
```

```
Enlace a [Rstudio](https://www.rstudio.com)
```

```
Imagen: ![Logo de RStudio](rstudio-Ball.png)
```

```
` Texto como código `
```

```
^[[Nota a pie de página]
```



Sintaxis R Markdown

Formato en línea

Texto en *cursiva* o cursiva

Texto en **negrita** o negrita

Subíndice₂

Superíndice²

Tachado

Enlace a RStudio



Imagen:

1

1. Nota a pie de página

Formateando el texto: sintaxis en bloque

```
# Encabezado 1
```

```
## Encabezado 2
```

Encabezado 1

Encabezado 2

> Cita textual

Cita textual

Tablas:

| Cabecera 1 | Cabecera 2 |
|------------|------------|
|------------|------------|

| Cabecera 1 | Cabecera 2 |
|------------|------------|
| ----- | ----- |

| | |
|---------|---------|
| Celda 1 | Celda 2 |
|---------|---------|

| | |
|---------|---------|
| Celda 3 | Celda 4 |
|---------|---------|

| Cabecera 1 | Cabecera 2 |
|------------|------------|
|------------|------------|

| | |
|---------|---------|
| Celda 1 | Celda 2 |
|---------|---------|

| | |
|---------|---------|
| Celda 3 | Celda 4 |
|---------|---------|

Formateando el texto: sintaxis en bloque

Listas desordenadas:

- * Elemento 1
- * Elemento 2
 - Sub-elemento 1
 - Sub-elemento 2

Listas ordenadas:

1. Elemento 1
2. Elemento 2
 - Sub-elemento 1
 - Sub-elemento 2

- Elemento 1
- Elemento 2

1. Elemento 1
2. Elemento 2

```
---
```

```
title: "Sintaxis R Markdown"
```

```
output: html_document
```

```
---
```

```
***
```

```
## Elementos en bloque
```

```
***
```

```
# Encabezado 1
```

```
## Encabezado 2
```

```
### Encabezado 3
```

```
* Lista desordenada
```

```
* Elemento 2
```

```
  + Sub-elemento 1
```

```
  - Sub-elemento 2
```

```
1. Lista ordenada
```

```
2. Elemento 2
```

```
  + Sub-elemento 1
```

```
  + Sub-elemento 2
```

```
> Cita
```



Sintaxis R Markdown

Elementos en bloque

Encabezado 1

Encabezado 2

Encabezado 3

- Lista desordenada
 - Elemento 2
 - Sub-elemento 1
 - Sub-elemento 2
1. Lista ordenada
 2. Elemento 2
 - Sub-elemento 1
 - Sub-elemento 2

Cita

Formateando el texto: expresiones matemáticas

- Ecuaciones en línea.

$$f(k) = \binom{m}{k} p^k (1-p)^{n-k}$$

$$f(k) = \binom{m}{k} p^k (1 - p)^{n-k}$$

- Ecuaciones aparte.

$$f(k) = \binom{m}{k} p^k (1-p)^{n-k}$$

$$f(k) = \binom{m}{k} p^k (1 - p)^{n-k}$$

- Entornos matemáticos.

```
$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}
```

```
$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc
```

$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

Apuntes LáTeX: Fórmulas matemáticas - Conceptos básicos

(<http://metodos.fam.cie.uva.es/~latex/curso-2015/apuntes3.pdf>)

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

```

---
title: "Sintaxis R Markdown"
output: html_document
---

***

## Tablas

***

Cabecera 1	Cabecera 2
Celda 1 | Celda 2
Celda 3 | Celda 4

***

## Expresiones matemáticas

***

Ecuaciones en línea:

$f(k) = {m \choose k} p^k (1-p)^{n-k}$

Ecuaciones aparte:

$$f(k) = {m \choose k} p^k (1-p)^{n-k}$$

Entornos matemáticos:

$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

```



Sintaxis R Markdown

Tablas

| Cabecera 1 | Cabecera 2 |
|------------|------------|
| Celda 1 | Celda 2 |
| Celda 3 | Celda 4 |

Expresiones matemáticas

Ecuaciones en línea:

$$f(k) = {m \choose k} p^k (1-p)^{n-k}$$

Ecuaciones aparte:

$$f(k) = {m \choose k} p^k (1-p)^{n-k}$$

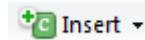
Entornos matemáticos:

$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

5. Trozos de código R (code chunks)

Existen **dos tipos de código R** en un documento R Markdown:

- Código **en línea**. Se escribe entre `r`.
- **Trozos de código**. Comienzan con ```{r nombre_trozo} y terminan con ```.

Se insertan rápidamente pulsando el botón 

Podemos añadir **etiquetas** para nombrar los trozos de código (deben ser únicas) y personalizar la salida del código con distintas **opciones** que se añaden como argumentos en la cabecera.

Ejemplo:

```
```{r qplot, fig.align = 'center', fig.width = 4, fig.height = 3, message = FALSE}
library(ggplot2)
summary(cars)
qplot(speed, dist, data = cars) + geom_smooth()
...``
```

**Asignar un nombre a cada trozo de código nos permite:**

- Tener identificados los chunks y poder navegar rápidamente por ellos.
  - Son importantes cuando guardamos los objetos creados por los trozos de código (almacenamiento en caché y nombre de las figuras).

# Trozos de código R: Opciones

Los trozos de código R pueden personalizarse con las opciones de [knitr](https://yihui.name/knitr/options/) (<https://yihui.name/knitr/options/>), especificando los argumentos dentro de los corchetes de la cabecera {r }:

Opción	Valor por defecto	Descripción
<b>echo</b>	TRUE	Mostrar código en el documento final
<b>eval</b>	TRUE	Evaluar trozo de código
<b>include</b>	TRUE	Si es FALSE no se incluye el código (pero si se evalúa) ni los resultados en el documento final
<b>warning</b>	TRUE	Mostrar advertencias generadas por el código
<b>error</b>	TRUE	Mostrar errores generados por el código
<b>message</b>	TRUE	Mostrar mensajes generados por el código
<b>fig.cap</b>	NULL	Vector de caracteres con leyenda para las figuras

# Trozos de código R: Más opciones

Opción	Valor por defecto	Descripción
<b>fig.height</b>	7	Altura de las figuras creadas por el trozo de código
<b>fig.width</b>	7	Anchura de las figuras creadas por el trozo de código
<b>fig.align</b>	'default'	Alineación de las figuras en el documento ('left', 'right', 'center')
<b>fig.path</b>	'figure/'	Directorio donde se guardarán las figuras creadas
<b>cache</b>	FALSE	Almacenamiento en caché de los resultados para reutilizarlos cada vez que se genere el documento (hasta que el código se altere)
<b>tidy</b>	FALSE	Visualizar el código de forma ordenada utilizando la función tidy_source() del paquete formatR (deja espacios entre operadores y las líneas largas de código las separa en varias líneas)
<b>comment</b>	'##'	Añadir una cadena de caracteres al comienzo de los resultados en el documento final
<b>results</b>	'markup'	'markup': mostrar los resultados con un cierto formato, 'hide': ocultar los resultados en el documento, 'hold': visualizar todos los resultados al final del trozo de código, 'asis': mostrar los resultados tal cual están en R

# 6. Opciones globales

Pueden establecerse opciones globales que se apliquen a todos los trozos de código del documento. Para ello utilizamos la función **opts\_chunk\$set()** del paquete knitr.

**Ejemplo:**

```
```{r knitr_options, include = FALSE}
knitr::opts_chunk$set(warning = FALSE, message = FALSE, echo = FALSE,
                      fig.path = 'Figuras/', fig.width = 12, fig.height = 4)
```

```

→ Nota: Utilizando :: podemos hacer uso de una función concreta, sin tener que cargar todo el contenido de un paquete.

# 7. Almacenamiento en caché

- En ocasiones, **la generación del documento puede llevar un tiempo considerable** cuando realizamos **cálculos largos** para obtener todos los resultados del informe.

Para mejorar el rendimiento del documento podemos **almacenar en caché** los fragmentos de código: `cache='TRUE'`. Éstos se guardarán la primera vez que se ejecute el código y, mientras no se realicen cambios, los objetos se cargarán directamente desde caché en las próximas generaciones del documento.

**Advertencia:** si un trozo de código no modificado depende de otros que sí, los objetos guardados en caché correspondientes al trozo no modificado no se actualizarán. Para evitar ésto utilizamos el argumento **dependson** en el que indicamos un vector de caracteres con las etiquetas de los trozos de código de los que depende o un vector numérico con el número correspondiente a los trozos de código (p.ej: `dependson=c(1,2)`). Este argumento sólo funciona en chunks locales, no en opciones globales.

- Por defecto, los resultados se guardan en una carpeta llamada `nombre_documento_cache` en el mismo directorio donde se encuentra nuestro documento. Podemos modificar este directorio y el nombre de la carpeta utilizando el argumento `cache.path = 'cache/...'`.

# 8. Tablas

Por defecto, R Markdown muestra los objetos de tipo `data.frame` y `matrix` como lo son en la terminal de R.

**Ejemplo:** El banco de datos `mtcars` contiene información sobre el consumo de combustible de 32 automóviles y otros aspectos relacionados con su diseño y rendimiento.

```
mtcars[1:6, 1:7]
```

```
mpg cyl disp hp drat wt qsec
Mazda RX4 21.0 6 160 110 3.90 2.620 16.46
Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02
Datsun 710 22.8 4 108 93 3.85 2.320 18.61
Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44
Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02
Valiant 18.1 6 225 105 2.76 3.460 20.22
```

Existen varios paquetes que permiten mostrar los datos con un formato adicional:

[xtable](https://cran.r-project.org/web/packages/xtable) (<https://cran.r-project.org/web/packages/xtable>), [stargazer](https://cran.r-project.org/web/packages/stargazer) (<https://cran.r-project.org/web/packages/stargazer>), [pander](https://cran.r-project.org/web/packages/pander) (<https://cran.r-project.org/web/packages/pander>), [knitr](https://cran.r-project.org/web/packages/knitr) (<https://cran.r-project.org/web/packages/knitr>), [huxtable](https://cran.r-project.org/web/packages/huxtable) (<https://cran.r-project.org/web/packages/huxtable>), [DT](https://cran.r-project.org/web/packages/DT) (<https://cran.r-project.org/web/packages/DT/>), [hwriter](https://cran.r-project.org/web/packages/hwriter) (<https://cran.r-project.org/web/packages/hwriter/>), [htmlTable](https://cran.r-project.org/web/packages/htmlTable) (<https://cran.r-project.org/web/packages/htmlTable>), ...

Debemos indicar el argumento `results='asis'` en el trozo de código que genera la tabla.

A continuación veremos algunos ejemplos.

# Tablas: knitr

Lo es utilizar la función `kable()` (<https://www.rdocumentation.org/packages/knitr/versions/1.16/topics/kable>) de . No tiene muchos argumentos pero en algunos casos es suficiente.

```
kable(x, format, digits =getOption("digits"), row.names = FALSE, col.names = NA, align,
caption = NULL, ...)
```

## Principales argumentos:

- x: un objeto matrix o data frame.
- format: "latex", "html", "markdown", "pandoc", "rst".
- digits: número de dígitos para las variables numéricas (puede ser un vector especificando valores para cada columna).
- row.names: valor lógico indicando si incluir el nombre de las filas.
- col.names: vector de caracteres indicando el nombre de las columnas.
- align: alineación de las columnas: 'l' (left), 'c' (center), 'r' (right).
- caption: título de la tabla.

# Ejemplo: kable()

```
knitr::kable(mtcars[1:6, 1:7], digits = 2, align = 'c', caption = "Banco de datos 'mtcars'.")
```

Banco de datos 'mtcars'.

|                          | mpg  | cyl | disp | hp  | drat | wt   | qsec  |
|--------------------------|------|-----|------|-----|------|------|-------|
| <b>Mazda RX4</b>         | 21.0 | 6   | 160  | 110 | 3.90 | 2.62 | 16.46 |
| <b>Mazda RX4 Wag</b>     | 21.0 | 6   | 160  | 110 | 3.90 | 2.88 | 17.02 |
| <b>Datsun 710</b>        | 22.8 | 4   | 108  | 93  | 3.85 | 2.32 | 18.61 |
| <b>Hornet 4 Drive</b>    | 21.4 | 6   | 258  | 110 | 3.08 | 3.21 | 19.44 |
| <b>Hornet Sportabout</b> | 18.7 | 8   | 360  | 175 | 3.15 | 3.44 | 17.02 |
| <b>Valiant</b>           | 18.1 | 6   | 225  | 105 | 2.76 | 3.46 | 20.22 |

# Tablas: pander

La función [pandoc.table\(\)](#)

(<https://www.rdocumentation.org/packages/pander/versions/0.6.0/topics/pandoc.table.return>) del paquete permite una mayor personalización.

```
pandoc.table(t, caption, decimal.mark = panderOptions("decimal.mark"), round = panderOptions("round"),
 missing = panderOptions("missing"), justify,
 style = c("multiline", "grid", "simple", "rmarkdown"), emphasize.strong.cols = c(1), ...)
```

## Principales argumentos:

- t: un objeto data frame, matrix o table.
- caption: título de la tabla.
- decimal.mark: símbolo decimal (";", ".").
- round: número de decimales en las variables numéricas.
- missing: cadena para reemplazar valores perdidos
- justify: 'left', 'centre', 'right', alineación de las celdas.
- style: estilo Pandoc: 'simple', 'multiline', 'grid' o 'rmarkdown'.
- emphasize.strong.rows, emphasize.strong.cols: filas y columnas en negrita.
- emphasize.italics.rows, emphasize.italics.cols: filas y columnas en cursiva.

El paquete incluye la función [panderOptions\(\)](#)

(<https://www.rdocumentation.org/packages/pander/versions/0.6.0/topics/panderOptions>) que permite establecer una gran cantidad de opciones.

# Ejemplo: pandoc.table()

```
pander::panderOptions('table.caption.prefix', "Tabla: ")
pander::pandoc.table(mtcars[1:6, 1:7], caption = "Banco de datos 'mtcars'.", justify = 'right',
 round = 2, emphasize.strong.cols = c(1, 5), emphasize.italics.rows = c(2, 3))
```

|                          | mpg         | cyl | disp | hp  | drat        | wt   | qsec  |
|--------------------------|-------------|-----|------|-----|-------------|------|-------|
| <b>Mazda RX4</b>         | <b>21</b>   | 6   | 160  | 110 | <b>3.9</b>  | 2.62 | 16.46 |
| <b>Mazda RX4 Wag</b>     |             |     |      |     |             |      |       |
| <b>Datsun 710</b>        |             |     |      |     |             |      |       |
| <b>Hornet 4 Drive</b>    | <b>21.4</b> | 6   | 258  | 110 | <b>3.08</b> | 3.21 | 19.44 |
| <b>Hornet Sportabout</b> | <b>18.7</b> | 8   | 360  | 175 | <b>3.15</b> | 3.44 | 17.02 |
| <b>Valiant</b>           | <b>18.1</b> | 6   | 225  | 105 | <b>2.76</b> | 3.46 | 20.22 |

---

Tabla: Banco de datos 'mtcars'.

# Tablas: xtable

La función `xtable()` (<https://www.rdocumentation.org/packages/xtable/versions/1.8-2/topics/xtable>) del paquete `xtable` permite también un control preciso de las tablas.

→ Si le damos una salida `lm()` produce automáticamente la tabla de coeficientes de regresión.

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

## Principales argumentos:

- `x`: objeto de R cuya clase aparezca en `methods(xtable)` (`xtable.matrix`, `xtable.data.frame`, `xtable.anova`, `xtable.lm`, `xtable.glm`, `xtable.prcomp`, ...).
- `caption`: título de la tabla.
- `align`: vector de caracteres indicando la alineación de las columnas.
- `digits`: número de dígitos a mostrar en las columnas.
- `display`: vector de caracteres indicando el formato de las columnas ("d": enteros, "f": reales, "s": cadenas, ...).
- `auto`: `TRUE/FALSE`, aplicar un formato automático cuando no se pase ningún valor a , o .

# Ejemplo: xtable()

Utilizando la función `print()` (<https://www.rdocumentation.org/packages/xtable/versions/1.8-2/topics/print.xtable>) junto con `xtable` podemos obtener una mayor personalización.

```
library(xtable)
model <- lm(mpg ~ ., data = mtcars)
print(xtable(model, align = "l|ccrr", digits = 2), type = "html")
```

|             | Estimate | Std. Error | t value | Pr(> t ) |
|-------------|----------|------------|---------|----------|
| (Intercept) | 12.30    | 18.72      | 0.66    | 0.52     |
| cyl         | -0.11    | 1.05       | -0.11   | 0.92     |
| disp        | 0.01     | 0.02       | 0.75    | 0.46     |
| hp          | -0.02    | 0.02       | -0.99   | 0.33     |
| drat        | 0.79     | 1.64       | 0.48    | 0.64     |
| wt          | -3.72    | 1.89       | -1.96   | 0.06     |
| qsec        | 0.82     | 0.73       | 1.12    | 0.27     |
| vs          | 0.32     | 2.10       | 0.15    | 0.88     |
| am          | 2.52     | 2.06       | 1.23    | 0.23     |
| gear        | 0.66     | 1.49       | 0.44    | 0.67     |
| carb        | -0.20    | 0.83       | -0.24   | 0.81     |

# Tablas: htmlTable

La función htmlTable()

(<https://www.rdocumentation.org/packages/htmlTable/versions/1.9/topics/htmlTable>) del paquete permite construir tablas avanzadas en HTML.

```
htmlTable(x, header, rnames, rowlabel, caption, tfoot, align = paste(rep("c", ncol(x)),
css.table = "margin-top: 1em; margin-bottom: 1em;", pos.caption, col.columns, ...)
```

## Principales argumentos:

- x: matrix o data frame con los datos.
- header: vector de caracteres especificando el nombre de las columnas.
- rnames: nombre de las filas.
- rowlabel: cabecera de la columna que contiene el nombre de las filas.
- caption: título de la tabla.
- tfoot: Añadir pie de tabla.
- align: alineación de las columnas ('l', 'c', 'r').
- css.table: especificar estilo de la tabla ("margin-top: 1em; margin-bottom: 1em;").
- pos.caption: posición del título ("bottom","top").
- col.columns: color de fondo de las columnas.

# Ejemplo: htmlTable()

```
htmlTable::htmlTable(round(mtcars[1:6, 1:8],2), col.columns = c(rep("#E6E6F0", 1),
rep("#F0FD99", 7)), css.cell = "padding: 0.5em")
```

|                   | mpg  | cyl | disp | hp  | drat | wt   | qsec  | vs |
|-------------------|------|-----|------|-----|------|------|-------|----|
| Mazda RX4         | 21   | 6   | 160  | 110 | 3.9  | 2.62 | 16.46 | 0  |
| Mazda RX4 Wag     | 21   | 6   | 160  | 110 | 3.9  | 2.88 | 17.02 | 0  |
| Datsun 710        | 22.8 | 4   | 108  | 93  | 3.85 | 2.32 | 18.61 | 1  |
| Hornet 4 Drive    | 21.4 | 6   | 258  | 110 | 3.08 | 3.21 | 19.44 | 1  |
| Hornet Sportabout | 18.7 | 8   | 360  | 175 | 3.15 | 3.44 | 17.02 | 0  |
| Valiant           | 18.1 | 6   | 225  | 105 | 2.76 | 3.46 | 20.22 | 1  |

## ► Comparación de las funciones de creación de tablas anteriores:

| Función | Lo mejor                                                                                             | Formatos de salida compatibles                               |
|---------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
|         | Sencillez y facilidad de uso                                                                         | Compatibilidad con todos los formatos (HTML, PDF, Word, ...) |
|         | Personalización avanzada                                                                             | Compatibilidad con todos los formatos (HTML, PDF, Word, ...) |
|         | Permite como argumento<br>objetos de tipo lm, glm, anova,<br>..., buena personalización de<br>tablas | Compatible sólo con PDF y HTML                               |
|         | Personalización avanzada de<br>tablas, especificación de estilos<br>mediante CSS                     | Compatible sólo con HTML                                     |

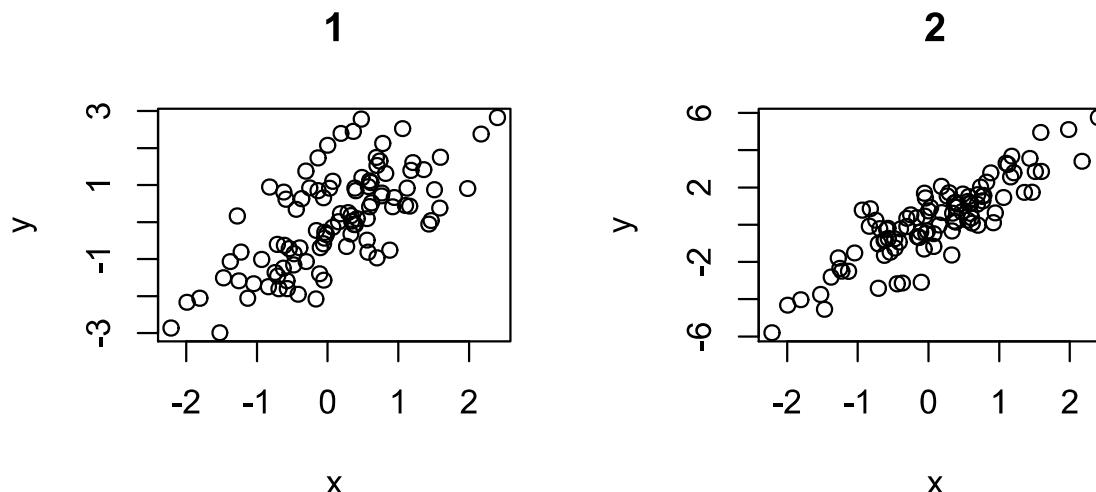
---

# 9. Figuras

La figuras producidas por los trozos de código se insertarán automáticamente en el documento final.

Ejemplo:

```
set.seed(1)
n <- 100
x <- rnorm(n)
par(mfrow = c(1, 2))
for (i in 1:2) {
 y <- i*x + rnorm(n)
 plot(x, y , main=i)
}
```



# Figuras: Algunas opciones

| Argumento                    | Descripción                                                                                                                                                                                                                                |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>fig.path</b>              | Directorio donde se guardarán las figuras creadas (p.ej: 'figures/').                                                                                                                                                                      |
| <b>fig.show</b>              | Opciones para mostrar las figuras ('asis': en el lugar exacto donde se generan, 'hold': todas las figuras al final del trozo, 'animate': envolver los gráficos en una animación, 'hide': generar gráficos pero ocultarlos en el documento) |
| <b>dev</b>                   | Nombre de la función que se utilizará como dispositivo gráfico para crear las figuras ('bmp', 'pdf', 'png', 'svg', 'jpeg', 'tiff', 'tikz', ...)                                                                                            |
| <b>dev.args</b>              | Otros argumentos que se pasarán al dispositivo gráfico (p.ej: dev.args=list(bg='yellow', pointsize=10)).                                                                                                                                   |
| <b>fig.ext</b>               | Extensión del fichero en la figura de salida.                                                                                                                                                                                              |
| <b>fig.width, fig.height</b> | Ancho y alto del gráfico en el dispositivo gráfico (7 por defecto).                                                                                                                                                                        |
| <b>out.width, out.height</b> | Ancho y alto del gráfico en el fichero de salida final (p.ej: '10cm').                                                                                                                                                                     |
| <b>fig.align</b>             | Alineación de las figuras en el documento ('left', 'right', 'center').                                                                                                                                                                     |
| <b>fig.cap</b>               | Vector de caracteres con leyenda para las figuras.                                                                                                                                                                                         |

# 10. Formatos de salida

Hay muchos [formatos de salida](http://rmarkdown.rstudio.com/formats.html) (<http://rmarkdown.rstudio.com/formats.html>) disponibles para usar con R Markdown.

- HTML ([http://rmarkdown.rstudio.com/html\\_document\\_format.html](http://rmarkdown.rstudio.com/html_document_format.html))
- PDF ([http://rmarkdown.rstudio.com/pdf\\_document\\_format.html](http://rmarkdown.rstudio.com/pdf_document_format.html))
- Word ([http://rmarkdown.rstudio.com/word\\_document\\_format.html](http://rmarkdown.rstudio.com/word_document_format.html)), ODT ([http://rmarkdown.rstudio.com/odt\\_document\\_format.html](http://rmarkdown.rstudio.com/odt_document_format.html))
- Presentaciones  
([http://rmarkdown.rstudio.com/beamer\\_presentation\\_format.html](http://rmarkdown.rstudio.com/beamer_presentation_format.html))
- Journals (<https://github.com/rstudio/rticles>)

El formato de salida se especifica en el campo **output** de la cabecera (YAML) del documento. Cada formato permite distintos argumentos con **opciones** para modificar el **estilo** y **estructura** del documento final.

# HTML

Para crear un **documento html** especificamos **output: html\_document** en la cabecera (YAML).

Personalizamos el documento de salida, añadiendo los argumentos necesarios como sub-campo del argumento output de la siguiente forma:

```

title: "Ejemplo HTML"
output:
 html_document:
 toc: true
 fig_width: 7
 fig_height: 6

```

→ Los opciones de cada campo deben aparecer tabuladas.

# HTML: Opciones

| Argumento                    | Descripción                                                                      |
|------------------------------|----------------------------------------------------------------------------------|
| <b>toc: true</b>             | Añadir tabla de contenidos al inicio del documento                               |
| <b>toc_depth: 2</b>          | Nivel de encabezado máximo que debe aparecer en la tabla de contenidos           |
| <b>toc_float: true</b>       | Aparezca la tabla de contenidos a la izquierda del documento final               |
| <b>number_sections: true</b> | Incluir numeración en los encabezados                                            |
| <b>theme: default</b>        | Utilizar temas predefinidos ("default", "cerulean", "journal", "flatly", ...).   |
| <b>highlight: default</b>    | Estilo de resaltado de la sintaxis ("default", "tango", "pygments", "kate", ...) |
| <b>fig_width, fig_height</b> | Establecer anchura y altura de las figuras                                       |
| <b>fig_caption: true</b>     | Representar figuras con subtítulos                                               |
| <b>df_print: kable</b>       | Visualización de tablas ("default", "kable", "tibble")                           |
| <b>keep_md: true</b>         | Guardar documento .md del proceso                                                |

# HTML: Estilo personalizado

Podemos añadir un **estilo propio** al documento HTML mediante CSS (<https://www.w3schools.com/css/>) especificando en la cabecera el archivo de estilos de la siguiente forma:

```

```

```
title: "Ejemplo HTML"
```

```
output:
```

```
 html_document:
```

```
 css: styles.css
```

```

```

## Ejemplo de archivo CSS (styles.css)

Para definir el **estilo de secciones específicas** incluimos un identificador en el encabezado de la sección:

```
Sección 3 {.resaltar}
```

# HTML: Inclusión de contenido HTML adicional

Podemos añadir contenido [HTML](https://www.w3schools.com/html/) (<https://www.w3schools.com/html/>) adicional en el documento de la siguiente forma:

```

title: "Ejemplo HTML"
output:
 html_document:
 includes:
 in_header: header.html
 before_body: doc_prefix.html
 after_body: doc_suffix.html

```

- ▶ **header.html** → Añadir información que no forma parte del contenido de la web: título, vínculos a hojas de estilo CSS, información para los robots de búsqueda, ...
- ▶ **doc\_prefix.html** → Insertar una cabecera antes del contenido de la web, logos, ...
- ▶ **doc\_suffix.html** → Añadir información al final de la página (dirección de contacto, estilo CSS utilizado, copyright, ...)

# PDF (LaTeX)

Para crear un **documento PDF** especificamos **output: pdf\_document** en la cabecera (YAML). Se requiere LaTeX en el equipo (se recomienda instalación completa).

Personalizamos el documento de salida, añadiendo los argumentos necesarios como sub-campo del argumento **output** de la siguiente forma:

```

title: "Ejemplo PDF"
output:
 pdf_document:
 toc: true
 number_sections: true
 keep_tex: true

```

# PDF: Opciones

| Argumento                      | Descripción                                                                                                                                                                                                                       |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>toc: true</b>               | Añadir tabla de contenidos al inicio del documento                                                                                                                                                                                |
| <b>toc_depth: 2</b>            | Nivel de encabezado mínimo que debe aparecer en la tabla de contenidos                                                                                                                                                            |
| <b>number_sections: true</b>   | Incluir numeración en los encabezados                                                                                                                                                                                             |
| <b>fig_width: 7</b>            | Establecer anchura predeterminada de las figuras                                                                                                                                                                                  |
| <b>fig_height: 6</b>           | Establecer altura predeterminada de las figuras                                                                                                                                                                                   |
| <b>fig_caption: true</b>       | Representar figuras con subtítulos                                                                                                                                                                                                |
| <b>df_print: kable</b>         | Visualización de tablas ("default", "kable", "tibble")                                                                                                                                                                            |
| <b>highlight: default</b>      | Estilo de resaltado de la sintaxis ("default", "tango", "pygments", "kate", ...)                                                                                                                                                  |
| <b>keep_tex: true</b>          | Guardar documento .tex del proceso                                                                                                                                                                                                |
| <b>template: plantilla.tex</b> | Plantilla Pandoc que se usará al generar el documento ( <a href="https://github.com/jgm/pandoc-templates">pandoc-templates</a> ( <a href="https://github.com/jgm/pandoc-templates">https://github.com/jgm/pandoc-templates</a> )) |

# PDF: Inclusión de contenido LaTeX adicional

Podemos añadir contenido LaTeX adicional en el documento de la siguiente forma:

```

title: "Ejemplo PDF"
output:
 pdf_document:
 includes:
 in_header: header.tex
 before_body: doc_prefix.tex
 after_body: doc_suffix.tex

```

- ▶ **header.tex** → Cargar paquetes de LaTeX, ...
- ▶ **doc\_prefix.tex** → Insertar una cabecera al documento, logos, ...
- ▶ **doc\_suffix.tex** → Información al final del documento (dirección de contacto, bibliografía, anexos...)

# PDF: Opciones LaTeX

| Argumento                                     | Descripción                                                                                              |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>lang</b>                                   | Lenguaje del documento                                                                                   |
| <b>fontsize</b>                               | Tamaño de fuente (P.ej: 11pt)                                                                            |
| <b>documentclass</b>                          | Tipo de documento (P.ej: article)                                                                        |
| <b>classoption</b>                            | Opciones del tipo de documento (P.ej: oneside)                                                           |
| <b>geometry</b>                               | Márgenes (P.ej: margin=1in)                                                                              |
| <b>mainfont, sansfont, monofont, mathfont</b> | Tipos de letra (únicamente funciona con <code>\usepackage{fontspec}</code> y <code>\setmainfont</code> ) |
| <b>linkcolor, urlcolor, citecolor</b>         | Color para links internos, externos y citas (red, green, magenta, cyan, blue, black)                     |

Estas opciones no aparecen como sub-campo de la sección output sino que aparecen en el nivel superior junto a title, author,...

# Word

Para crear un **documento word** especificamos output: word\_document en la cabecera (YAML).

Personalizamos el documento de salida, añadiendo los argumentos necesarios como sub-campo del argumento output de la siguiente forma:

```

title: "Ejemplo Word"
output:
 word_document:
 highlight: tango
 df_print: kable
 fig_caption: true

```

# Word: Opciones

| Argumento                            | Descripción                                                                                                                                                                   |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>fig_width: 7</b>                  | Anchura predeterminada de las figuras                                                                                                                                         |
| <b>fig_height: 6</b>                 | Altura predeterminada de las figuras                                                                                                                                          |
| <b>fig_caption: true</b>             | Representar figuras con subtítulos                                                                                                                                            |
| <b>df_print: kable</b>               | Visualización de tablas ("default", "kable", "tibble")                                                                                                                        |
| <b>highlight: default</b>            | Estilo de resaltado de la sintaxis ("default", "tango", "pygments", "kate", ...)                                                                                              |
| <b>keep_md: true</b>                 | Guardar documento .md del proceso                                                                                                                                             |
| <b>reference_docx: mystyles.docx</b> | Especificar estilo según un documento word de referencia<br>( <a href="http://rmarkdown.rstudio.com/articles_docx.html">http://rmarkdown.rstudio.com/articles_docx.html</a> ) |

# Presentaciones

R Markdown permite cuatro formatos de presentación:

- beamer\_presentation  
([http://rmarkdown.rstudio.com/beamer\\_presentation\\_format.html](http://rmarkdown.rstudio.com/beamer_presentation_format.html))  
presentaciones PDF con beamer
- ioslides\_presentation  
([http://rmarkdown.rstudio.com/ioslides\\_presentation\\_format.html](http://rmarkdown.rstudio.com/ioslides_presentation_format.html))  
presentaciones HTML con ioslides
- slidy\_presentation  
([http://rmarkdown.rstudio.com/slidy\\_presentation\\_format.html](http://rmarkdown.rstudio.com/slidy_presentation_format.html)) presentaciones HTML con slide
- revealjs\_presentation  
([http://rmarkdown.rstudio.com/revealjs\\_presentation\\_format.html](http://rmarkdown.rstudio.com/revealjs_presentation_format.html))  
presentaciones HTML con reveal.js

Para crear **nuevas diapositivas** utilizamos las etiquetas de encabezado # y ##.

Para crear **diapositivas sin cabecera** utilizamos ----.

# Ejemplo: slidy\_presentation

```

```

```
title: "My title"
author: "My name"
date: "Junio de 2017"
output: slidy_presentation

```

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
````
```

## ## R Markdown

This is an R Markdown presentation. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <<http://rmarkdown.rstudio.com>>.

when you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

## ## slide with Bullets

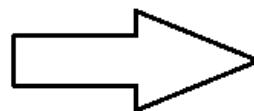
- Bullet 1
- Bullet 2
- Bullet 3

## ## slide with R Output

```
```{r cars, echo = TRUE}
summary(cars)
````
```

## ## slide with Plot

```
```{r pressure, fig.height=3, fig.width=3}
plot(pressure)
````
```



# My title

My name

Junio de 2017

Contents

slide 1/5

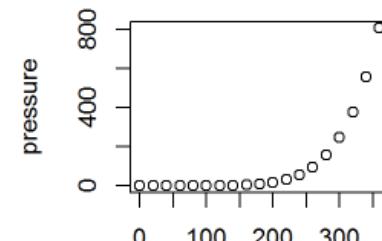
## Slide with Bullets

- Bullet 1
- Bullet 2
- Bullet 3

Contents

slide 3/5

## Slide with Plot



# R Markdown

This is an R Markdown presentation.

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.

Contents

slide 2/5

## Slide with R Output

```
summary(cars)
```

```
speed dist
Min. : 4.0 Min. : 2.00
1st Qu.:12.0 1st Qu.: 26.00
Median :15.0 Median : 36.00
Mean :15.4 Mean : 42.98
3rd Qu.:19.0 3rd Qu.: 56.00
Max. :25.0 Max. :120.00
```

Contents

slide 4/5

# Presentaciones: Algunas opciones

| Argumento                                    | Descripción                                                                      |
|----------------------------------------------|----------------------------------------------------------------------------------|
| <b>incremental: true</b>                     | Mostrar ítem de listas de forma incremental                                      |
| <b>highlight: default</b>                    | Estilo de resaltado de la sintaxis ("default", "tango", "pygments", "kate", ...) |
| <b>font_adjustment: -1</b>                   | Aumentar o disminuir el tamaño por defecto del texto                             |
| <b>css: styles.css</b>                       | Añadir un estilo propio mediante CSS                                             |
| <b>logo: logo.png</b>                        | Ruta a un archivo con un logotipo para añadirlo a la portada de la presentación  |
| <b>footer: "Copyright (c) 2014, RStudio"</b> | Añadir texto a pie de página                                                     |
| <b>fig_width: 7, fig_height: 6</b>           | Anchura y altura predeterminada de las figuras                                   |
| <b>fig_caption: true</b>                     | Representar figuras con subtítulos                                               |
| <b>df_print: kable</b>                       | Visualización de tablas ("default", "kable", "tibble")                           |
| <b>keep_md: true</b>                         | Guardar documento .md del proceso                                                |

# Journals

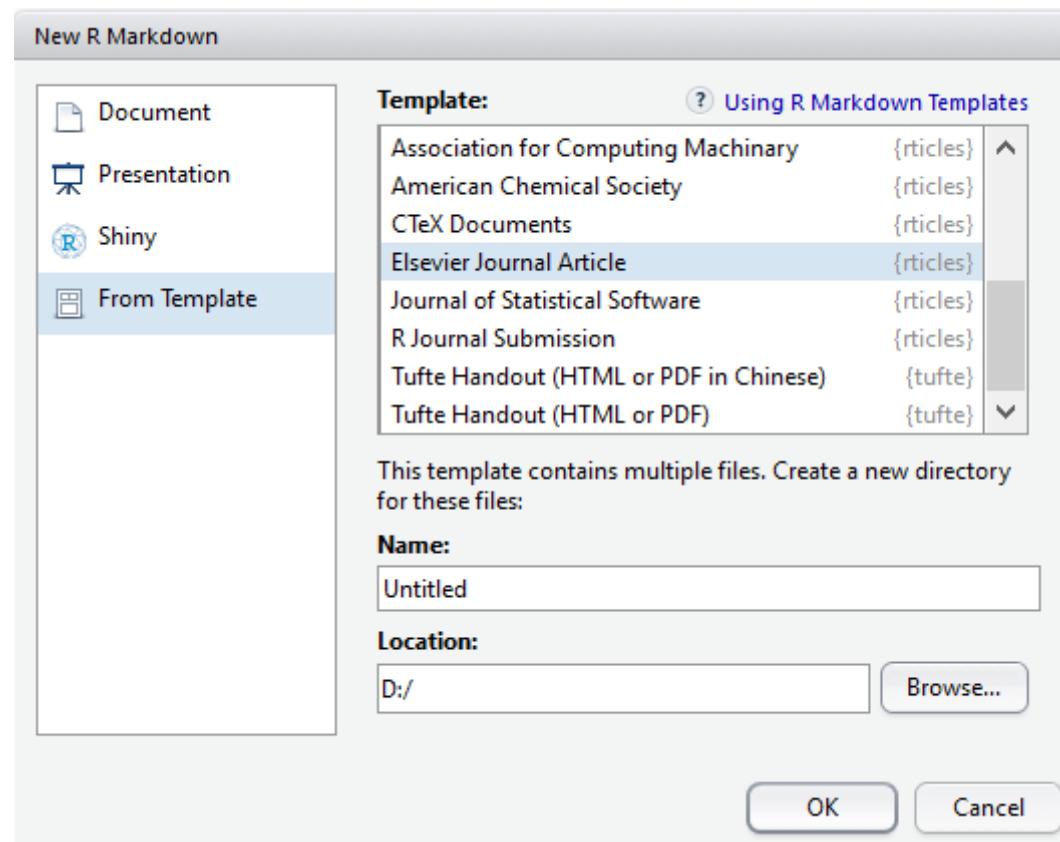
El paquete **rticles** proporciona un conjunto de plantillas de R Markdown-LaTeX para varios formatos:

- Artículos Journal of Statistical Software (<https://www.jstatsoft.org/index>)
- Artículos R Journal (<https://journal.r-project.org/>)
- Artículos Association for Computing Machinery (<http://www.acm.org/>)
- Artículos American Meteorological Society (<https://www.ametsoc.org/ams/>)
- Presentaciones a revistas de Elsevier (<https://www.elsevier.com/>)
- Presentaciones a revistas de American Economic Association (<https://www.aeaweb.org/journals/policies/templates>)

Estas plantillas aseguran que los documentos se ajustan exactamente a los estándares de envío.

# rticles desde RStudio: Requisitos

1. Instalar la última versión de RStudio (<https://www.rstudio.com/products/rstudio/download2/>).
2. Instalar el paquete **rticles**: `install.packages("rticles", type = "source")`
3. Usar el diálogo de R Markdown para crear un artículo desde uno de los templates:





# 11. Informes parametrizados

Los documentos R Markdown parametrizados son aquellos en los que se incluyen algunos **parámetros** y los resultados presentados en el informe van a depender del valor que asignemos a dichos parámetros. Estos informes son útiles cuando queremos **presentar el mismo análisis pero con distintos valores** (p.ej: generar un informe para distintos departamentos o regiones geográficas).

Los parámetros se declaran en el campo **params** de la cabecera del documento R Markdown. **Ejemplo:**

```

```

```
title: Probando informes con parámetros
output: html_document
params:
 ciudad: "Valencia, España"

```

En el código, accederemos a los valores de los parámetros escribiendo: **params\$ciudad**.

Para volver a generar el documento cambiando los valores de los parámetros utilizamos el argumento **params** de la función **render()**:

```
rmarkdown::render("Informe_ciudades.Rmd", params = list(ciudad = 'Alicante, España'))
```

# Ejemplo: Informes parametrizados

```

```

```
title: "Probando informes con parámetros"
output: html_document
params:
 ciudad: "valencia, España"

```

```
```{r options_global, include = FALSE}
knitr::opts_chunk$set(warning = FALSE, message = FALSE,
echo = FALSE)
````
```

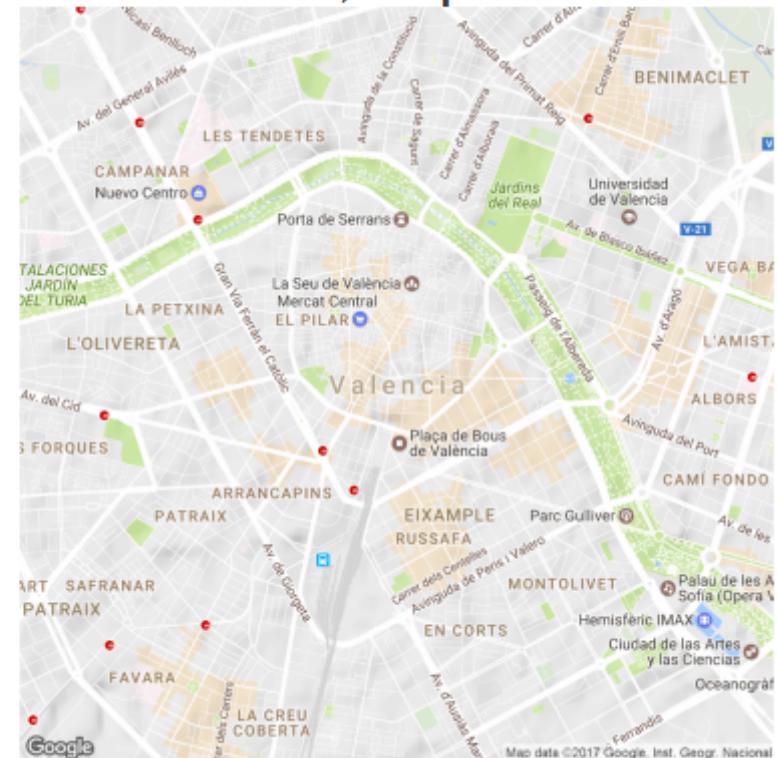
```
```{r qmap, results='asis'}
library(ggmap)
cat(paste0('## Mapa de ', params$ciudad, ':'))
```

```
qmap(params$ciudad, zoom = 14)
```



Probando informes con parámetros

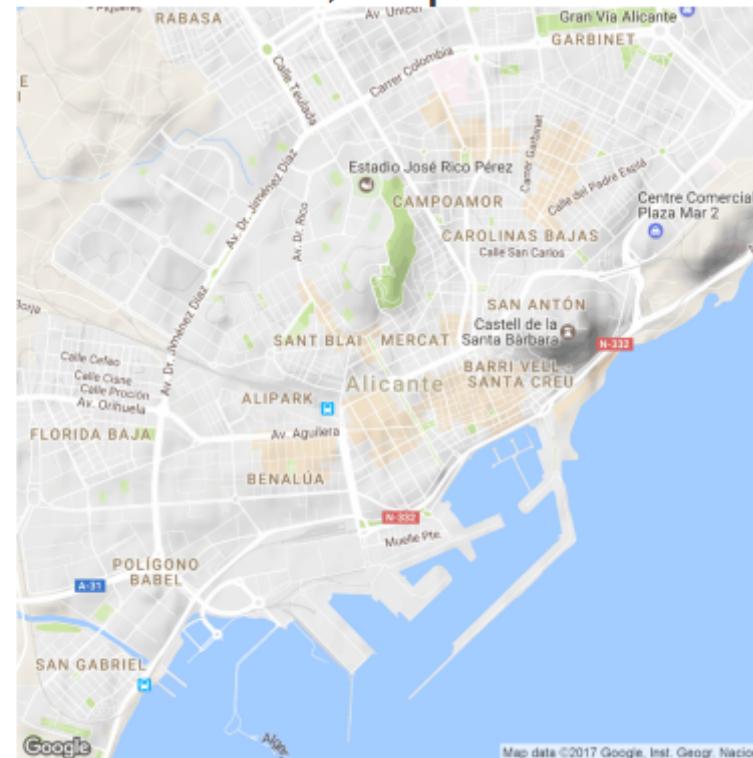
Mapa de Valencia, España:



```
rmarkdown::render("Informe_ciudades.Rmd", params = list(ciudad = 'Alicante, España'))
```

Probando informes con parámetros

Mapa de Alicante, España:



12. Bibliografía y citas

Podemos añadir fácilmente la bibliografía de nuestro documento especificando un archivo de bibliografía en el campo `bibliography` de la cabecera:

```
---
```

```
title: "Sample Document"
output: html_document
bibliography: bibliography.bib
---
```

Pandoc (<http://pandoc.org/MANUAL.html>) genera automáticamente la bibliografía y se coloca al final del documento.

Formatos de bibliografía permitidos:

Formato	Extensión del fichero
MODS	.mods
BibLaTeX	.bib
BibTeX	.bibtex
RIS	.ris
EndNote	.enl
EndNote XML	.xml
ISI	.wos
MEDLINE	.medline
Copac	.copac
JSON citeproc	.json

Citas

Las citas deben insertarse en el documento mediante una clave, formada por '@' + el identificador de la cita en la base de datos. Ejemplos:

► **Citas entre paréntesis:**

[@Gandrud2015; @Xie2015] → (Gandrud 2015; Xie 2015)

► **Citas sin paréntesis para los autores:**

@Gandrud2015 → Gandrud (2015)

► **Citas sin mencionar al autor:**

[-@Gandrud2015] → (2015)

Referencias no utilizadas

Para incluir artículos en la bibliografía que no son citados en el texto, añadimos en la cabecera del documento:

```
---
title: "Sample Document"
output: html_document
bibliography: bibliography.bib
nocite: |
  @Broman2016, @Peng2015, @Xie2016
---
```

Estilo de las citas

Por defecto, Pandoc utiliza el formato autor-fecha para citas y referencias. Para utilizar otro estilo, hay que especificar un archivo CSL en el campo `csl` de la cabecera. Ejemplo:

```
---
title: "Sample Document"
output: html_document
bibliography: bibliography.bib
csl: biomed-central.csl
---
```

- **Repositorio de estilos CSL:** <https://github.com/citation-style-language/styles>
[\(https://github.com/citation-style-language/styles\)](https://github.com/citation-style-language/styles)
- **Introducción a la creación y modificación de estilos CSL:**
<http://citationstyles.org/downloads/primer.html>
[\(http://citationstyles.org/downloads/primer.html\)](http://citationstyles.org/downloads/primer.html)

¿Qué nos permite R Markdown?

- **Texto, código y resultados en un mismo documento** lo que nos ayuda a conseguir que nuestros informes sean reproducibles (sabiendo exactamente cómo hemos obtenido los resultados).
- **Forma muy sencilla de dar formato al texto** de nuestro informe mediante el lenguaje .
- **Generación de tablas y figuras automáticamente** en el documento, evitando tener que escribir e insertar manualmente los resultados.
- **Almacenamiento automático de las figuras** creadas por los trozos de código en el directorio que elijamos.
- **Gran variedad de formatos de presentación de informes:** HTML, Word, PDF, Presentaciones, Artículos, ... sin tener que preocuparnos de maquetar manualmente el documento.
- **Generación automática del mismo informe para distintos datos** (informes parametrizados).
- **Incorporación automática de bibliografías** y sencillez en la citación de documentos.

Recursos útiles:

- R Markdown desde RStudio (<http://rmarkdown.rstudio.com/>).
- El R Markdown Cheatsheet (<https://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf>).
- La guía de referencia de R Markdown (<https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>).
- RPubs: "Creating Dynamic Documents with RMarkdown and Knitr" (<https://rpubs.com/marschmi/RMarkdown>).
- DataCamp: Curso "Reporting with R Markdown" (<https://www.datacamp.com/courses/reporting-with-r-markdown>).

Referencias

- Broman, Karl. 2016. "Writing Reproducible Reports: Knitr with R Markdown." <http://kbroman.org/Tools4RR/> (<http://kbroman.org/Tools4RR/>).
- Gandrud, Christopher. 2015. . Chapman; Hall/CRC.
- Peng, Roger D. 2015. "Markdown and R."
- Xie, Yihui. 2015. . Chapman; Hall/CRC.
- . 2016. . Chapman; Hall/CRC.
<https://bookdown.org/yihui/bookdown/> (<https://bookdown.org/yihui/bookdown/>).

¡ Muchas Gracias Por Vuestra Atención !



[\(https://creativecommons.org/licenses/by/4.0/deed.es_ES\)](https://creativecommons.org/licenses/by/4.0/deed.es_ES)

Este trabajo está bajo una [licencia de Creative Commons Reconocimiento 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/deed.es_ES) (https://creativecommons.org/licenses/by/4.0/deed.es_ES).

Práctica 1: R Markdown

- Crear un informe utilizando R Markdown.
- Banco de datos y una serie de tareas como guía para la creación del informe.
- Descargar el archivo : banco de datos y un objeto espacial
- Construir un índice de privación a partir de las variables del banco de datos mediante un análisis de componentes principales.
- Representar el índice de privación sobre la cartografía de la ciudad de Valencia.
- Presentar los resultados obtenidos en el informe.

