

# Fastbook 08

## Programación en R

Informes R Markdown



Edix Educación

## 08. Informes R Markdown

Como ya hemos comentado anteriormente, la última parte de la asignatura aborda los informes R Markdown. En este fastbook nos introduciremos en la construcción básica de este tipo de documentos mientras que, en el siguiente, cubriremos aspectos más avanzados.

¡Comencemos!

*Autor: Juan Jiménez García*

[Introducción a R Markdown](#)

[Estructura y funcionamiento](#)

[Markdown](#)

[Conclusiones](#)

# Introducción a R Markdown

**X** Edix Educación

---

Llegados a este punto, resulta muy interesante echar la vista atrás para reflexionar acerca del objetivo de esta asignatura y de la práctica de analytics en general.

Si recordáis, en el primer apartado del fastbook 01, citábamos la frase del famoso escritor Stephen Few.



“Los números tienen una historia importante que contar. Ellos confían en ti para que les des una voz.”

- Stephen Few

Como ya os dije en su momento, esta frase me gusta porque representa muy bien **cuál es la labor de un data scientist**. Los datos no tienen valor por sí mismos, lo realmente valioso es el **conocimiento que podemos extraer** a partir de ellos.

Para que dicho conocimiento sea obtenido de forma ideal, se tienen que llevar a cabo dos labores principales:

- La primera es explotar el dato de manera inteligente por medio de análisis, visualizaciones y modelos para obtener información relevante y de interés.
- El segundo paso, pero no por ello menos importante que el anterior, consiste en presentar, comunicar y transmitir dichos aprendizajes de una forma eficiente.

Como ya sabéis, en esta asignatura no nos hemos centrado en cómo debemos analizar, visualizar y modelar la información (esos aspectos ya se cubren en otras materias). **Nuestro objetivo** ha sido el de aprender el lenguaje de programación R como facilitador, herramienta y medio para poder llevar a cabo dichas tareas con fluidez.

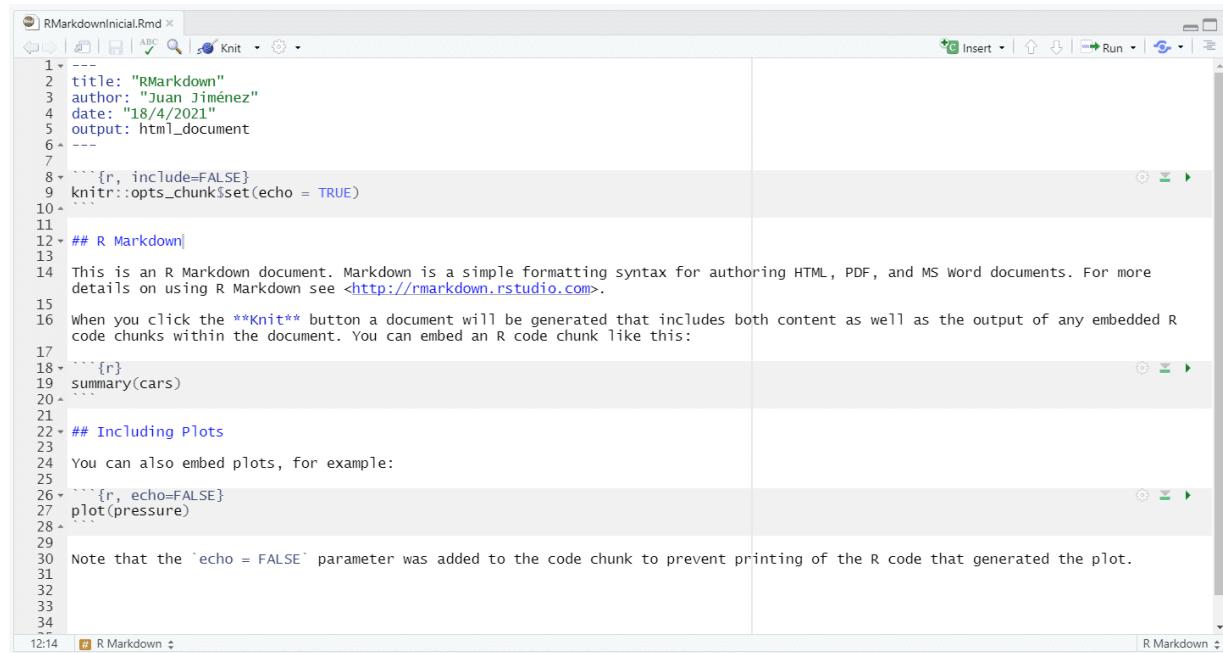
Continuando con esa misma lógica, en esta parte de la asignatura vamos a abordar R Markdown como herramienta y medio para poder comunicar nuestros resultados.

---

Los ficheros R Markdown nos permiten crear documentos de diferentes extensiones combinando texto formateado y código R. Su extensión es

*.Rmd*

La siguiente imagen muestra un fichero ejemplo R Markdown que está siendo editado en RStudio.



```
1 ---  
2 title: "RMarkdown"  
3 author: "Juan Jiménez"  
4 date: "18/4/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r, 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}  
19 summary(cars)  
20   
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r, 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  
32  
33  
34
```

Para su creación, desde RStudio debemos hacer click en *File / New File / R Markdown*.

La primera vez que lo hagamos se nos pedirá confirmación para la instalación de ciertos paquetes que resultan necesarios para el trabajo con este tipo de ficheros (aceptamos y, tras pocos segundos, el proceso habrá terminado).

# Estructura y funcionamiento

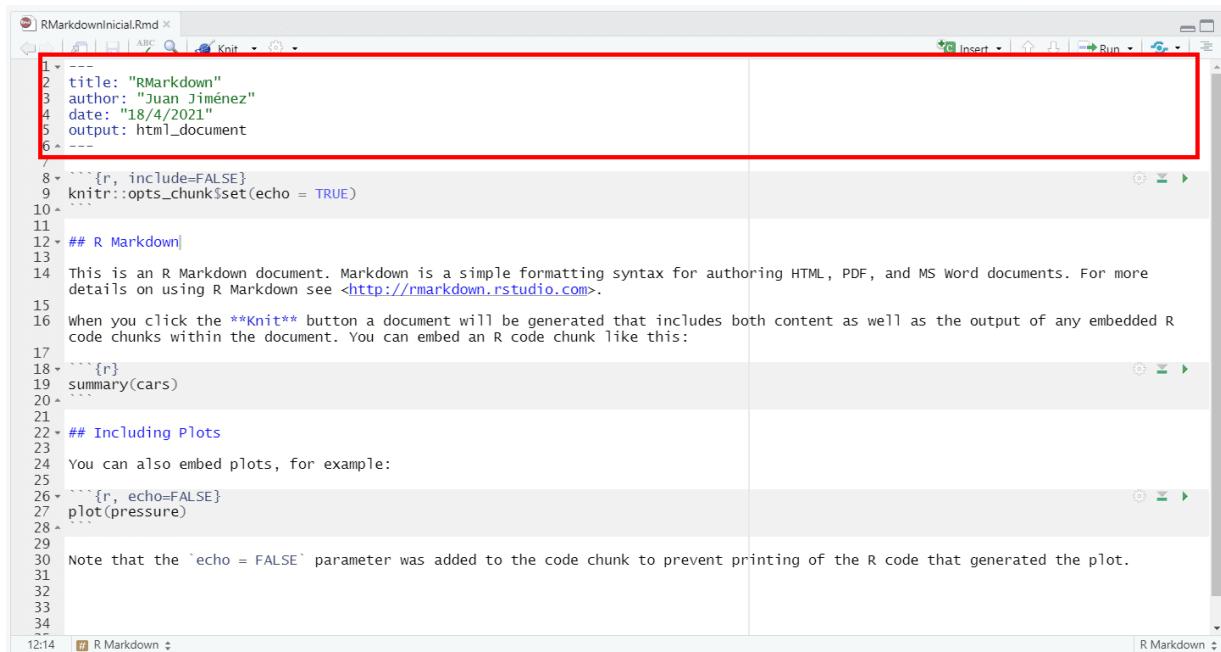
X Edix Educación

Los ficheros R Markdown contienen **tres tipos de contenido**:

1

YAML Header

Se encuentra al inicio del documento y está encerrado por tres guiones seguidos. En él se definen algunos **parámetros generales de formato** que estructuran el producto final. Los más utilizados y básicos son los que se muestran en la imagen: título, autor, fecha y formato de salida.



The screenshot shows an RStudio interface with an R Markdown file named "RMarkdownInicial.Rmd". The code editor displays the following content:

```
1 ---  
2 title: "RMarkdown"  
3 author: "Juan Jiménez"  
4 date: "18/4/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r, 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}  
19 summary(cars)  
20 ```  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r, 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  
32  
33  
34
```

A red box highlights the first six lines of the code, which define the YAML header.

2

## Chunks de código R

En esta zona del archivo podemos introducir cualquier tipo de **código R**, como si de un script se tratase (importación de paquetes, operaciones, gráficos, etc.). Tal y como se observa en la imagen, los textos ``{r} y `` indican, respectivamente, el inicio y el final del chunk.

```
RMarkdownInitial.Rmd ✘
1 ---  
2 title: "RMarkdown"  
3 author: "Juan Jiménez"  
4 date: "18/4/2021"  
5 output: html_document  
6 ---  
7  
8 ``{r, 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}  
19 summary(cars)  
20 ...  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ``{r, 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  
32  
33  
34  
12:14 R Markdown
```

3

## Texto formateado con Markdown

En esta parte se incluye el **texto** que acompaña a los análisis y cuyo formato y apariencia puede ser customizado con gran libertad gracias a Markdown (títulos, listados, negritas, etc.). En el siguiente apartado del fastbook aprenderemos sus reglas y utilidades.

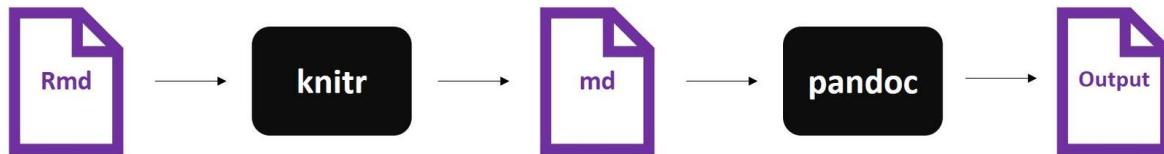
```

1 <!--
2 title: "RMarkdown"
3 author: "Juan Jiménez"
4 date: "18/4/2021"
5 output: html_document
6 ---
7
8 ````{r, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10
11 ## R Markdown
12
13 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more
14 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
17 code chunks within the document. You can embed an R code chunk like this:
18 ````{r}
19 summary(cars)
20
21 ## Including Plots
22
23 You can also embed plots, for example:
24 ````{r, echo=FALSE}
25 plot(pressure)
26
27 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
28
29
30
31
32
33
34
12:14 R Markdown
  
```

Una vez vistos los tres tipos de contenidos que albergan nuestros archivos R Markdown (cabecera, código R y texto formateado), es hora de entender su **proceso de exportación**.

Este procedimiento se lleva a cabo con la función **render()** (introduciendo dentro del paréntesis el nombre del archivo y el formato de salida) o con el botón de **RStudio Knit** (es la opción que yo os recomiendo, dado que previsualiza el resultado de forma automática).

La siguiente imagen nos muestra qué es lo que ocurre internamente al ejecutar la acción de exportación:



Nuestro archivo Rmd es recibido por el paquete knitr, quien ejecuta los chunks de código y construye un nuevo archivo md (markdown). Este contiene tanto el texto formateado incluido en el rmd como el código R que se ejecuta y los resultados que genera. Por último, el paquete pandoc convierte el markdown en el formato de salida que se haya seleccionado.

Podemos exportar a una gran cantidad de **formatos** que se pueden clasificar en dos tipos, documentos y presentaciones. En negrita se destacan aquellos más importantes:

## Documentos

---

- Word
- HTML
- PDF
- Markdown
- Open Document Text
- Rich Text Format

## Presentaciones

---

- Powerpoint
- Slidy
- Beamer
- Ioslides

También es importante hablar de las opciones que afectan a los chunks de código R y que nos permiten **configurar su visualización**. En concreto, existen dos tipos de opciones:

## 1

### Opciones locales

Las opciones locales se introducen **dentro de los corchetes** que marcan el inicio del chunk y afectan única y exclusivamente a este.

- ***include***: si lo asignamos a FALSE, el código del chunk y los resultados que genere no aparecerán en la salida, aunque sí que serán ejecutados. Es decir, si construimos una nueva variable en él, esta podrá ser usada en los siguientes chunks.
- ***echo***: si lo asignamos a FALSE, evitamos que el código aparezca en la salida, aunque sí lo harán los resultados de la ejecución (por ejemplo, un gráfico).
- ***message***: si lo asignamos a FALSE, evitamos que los mensajes generados por el código aparezcan en la salida.
- ***warning***: si lo asignamos a FALSE, evitamos que los warnings generados por el código aparezcan en la salida.
- ***fig.cap = “...”*** Nos permite añadir pie de foto a los resultados gráficos del chunk.

## 2

### Opciones globales

La función ***knitr::opts\_chunk\$set()*** nos permite modificar las opciones (*echo*, *warning*, etc.) para que se apliquen en todos los chunks del fichero y nos ahorremos su escritura de forma repetitiva.

Las opciones locales tienen más importancia que las globales,  
por lo que, en caso de contradicción en un chunk, manda la  
opción local.

La siguiente imagen ejemplifica el uso de algunas de las opciones locales y globales que acabamos de comentar.

```

1 ---  

2 title: "RMarkdown"  

3 author: "Juan Jiménez"  

4 date: "18/4/2021"  

5 output: html_document  

6 ---  

7 Local  

8 ```{r, include=FALSE}  

9 knitr::opts_chunk$set(echo = TRUE)  

10 ...  

11 Global  

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 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:  

16  

17```{r}  

18 summary(cars)  

19 ...  

20 Local  

21 ## Including Plots  

22  

23 You can also embed plots, for example:  

24  

25```{r, echo=FALSE}  

26 plot(pressure)  

27 ...  

28 Local  

29 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.  

30  

31  

32  

33  

34

```



[Programación en R, fb. 09: R markdown \(I\) - Edix](#)

# Markdown

X Edix Educación

---

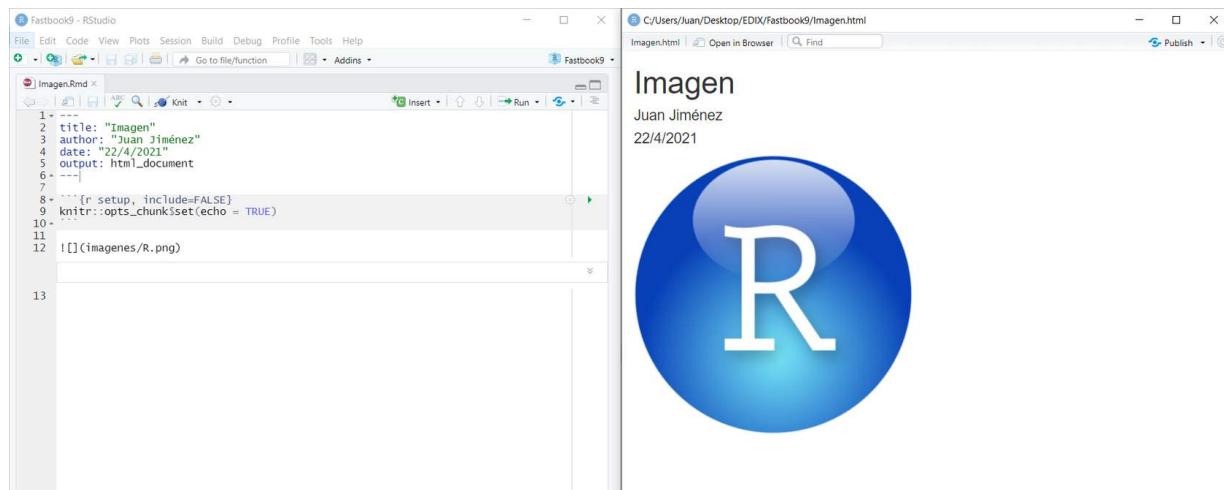
Hasta el momento, no hemos visto ninguna manera de introducir el formato o la apariencia en el texto de nuestros reportes. Sin embargo, tal y como os he adelantado, Markdown nos abre esa posibilidad. Empecemos por el principio.

---

Markdown es un lenguaje que permite aplicar formato a un texto gracias al uso de una serie de caracteres que especifican la apariencia a utilizar.

Nació como herramienta de conversión de texto plano a HTML (el lenguaje de marcado que utilizan las páginas web).

En la siguiente imagen podéis observar un archivo markdown (a la izquierda) y su visualización una vez transformado a HTML (a la derecha).



La transformación la realiza un conversor de formatos que entiende la sintaxis markdown y la transforma en sintaxis HTML. En nuestro caso, como ya hemos comentado, esta labor es realizada por el paquete de R **pandoc**.

Entonces..., ¿qué necesitamos aprender para poder usarlo?

---

**Para utilizar Markdown simplemente debemos conocer su  
sintaxis. Es realmente sencilla, así que vamos a ello.**

---

---

**Vamos a organizar el estudio en dos bloques. En el primero se presenta la sintaxis más esencial e importante, mientras que en el segundo se exponen aspectos más avanzados y concretos.**

## **Encabezados**

El primer elemento que vamos a aprender son los encabezados. Para ello debemos usar la almohadilla (#). Tenemos diferentes niveles y tamaños en función del número de almohadillas que insertemos delante del texto. **A más almohadillas, menor será el tamaño del título.**

En la siguiente tabla podemos observar el texto que se incluye dentro del fichero markdown (a la izquierda) y cómo se visualiza cuando lo renderizamos y convertimos en HTML (a la derecha).

Markdown	Visualización
# Heading nivel 1	<b>Heading nivel 1</b>
## Heading nivel 2	<b>Heading nivel 2</b>
### Heading nivel 3	<b>Heading nivel 3</b>
#### Heading nivel 4	<b>Heading nivel 4</b>
##### Heading nivel 5	<b>Heading nivel 5</b>
###### Heading nivel 6	<b>Heading nivel 6</b>

## Negrita

Para hacer que un texto o palabra aparezca en negrita debemos encerrarlo entre dos pares de asteriscos (\*\*).

La siguiente tabla muestra un par de ejemplos.

Markdown	Visualización
Usamos **dos asteriscos** para **aplicar formato bold** a nuestro texto	Usamos dos asteriscos para aplicar formato bold a nuestro texto
Usamos la **negrita** para **destacar contenido de nuestro texto**	Usamos la negrita para destacar contenido de nuestro texto

## Cursiva

Para hacer que un texto o palabra aparezca en cursiva debemos encerrarlo entre dos asteriscos (\*). La siguiente tabla muestra un par de ejemplos.

Markdown	Visualización
Usamos <b>*un asterisco*</b> para <b>*aplicar formato italic*</b> a nuestro texto	Usamos <i>un asterisco</i> para <i>aplicar formato italic</i> a nuestro texto
Usamos la <b>*cursiva*</b> para <b>*introducir énfasis en nuestro texto*</b>	Usamos <i>la cursiva</i> para <i>introducir énfasis en nuestro texto</i>

## Negrita y cursiva

Para hacer que un texto o palabra aparezca en negrita y cursiva, debemos encerrarlo entre dos tríos de asteriscos (\*\*\*) . La siguiente tabla muestra un par de ejemplos.

Markdown	Visualización
Usamos <b>***tres asteriscos***</b> para <b>***aplicar formato italic y bold***</b> a nuestro texto	Usamos <i>tres asteriscos</i> para <i>aplicar formato italic y bold</i> a nuestro texto
Usamos la <b>***negrita y cursiva***</b> para <b>***destacar y dar énfasis a cierto contenido de nuestro texto***</b>	Usamos la <i>negrita y cursiva</i> para <i>destacar y dar énfasis a cierto contenido de nuestro texto</i>

## Lista con orden

Markdown también nos permite construir listados (en este caso numéricos) tal y como lo haríamos en Word o cualquier editor de texto.

La siguiente tabla muestra diferentes ejemplos. Fijaos bien en qué es lo que ocurre cuando no construimos la lista con el orden numérico incorrecto (respeta el inicio, pero luego va de uno en uno obligatoriamente).

Markdown	Visualización
<pre> 1. Primer elemento 2. Segundo elemento 3. Tercer elemento 4. Cuarto elemento   1. Primer elemento sublistा   2. Segundo elemento sublistা  10. Décimo elemento 11. Undécimo elemento 16. Duodécimo elemento  1. Décimo elemento 2. Undécimo elemento 5. Duodécimo elemento </pre>	<ul style="list-style-type: none"> <li>1. Primer elemento</li> <li>2. Segundo elemento</li> <li>3. Tercer elemento</li> <li>4. Cuarto elemento           <ul style="list-style-type: none"> <li>1. Primer elemento sublistা</li> <li>2. Segundo elemento sublistা</li> </ul> </li>   <li>10. Décimo elemento</li> <li>11. Undécimo elemento</li> <li>12. Duodécimo elemento</li>   <li>1. Décimo elemento</li> <li>2. Undécimo elemento</li> <li>3. Duodécimo elemento</li> </ul>

## Listas sin orden

De igual forma que en el caso anterior, podemos construir listas no numéricas (sin orden). Para ello, tal y como se observa en el siguiente ejemplo, podemos usar el guion, el asterisco o la suma.

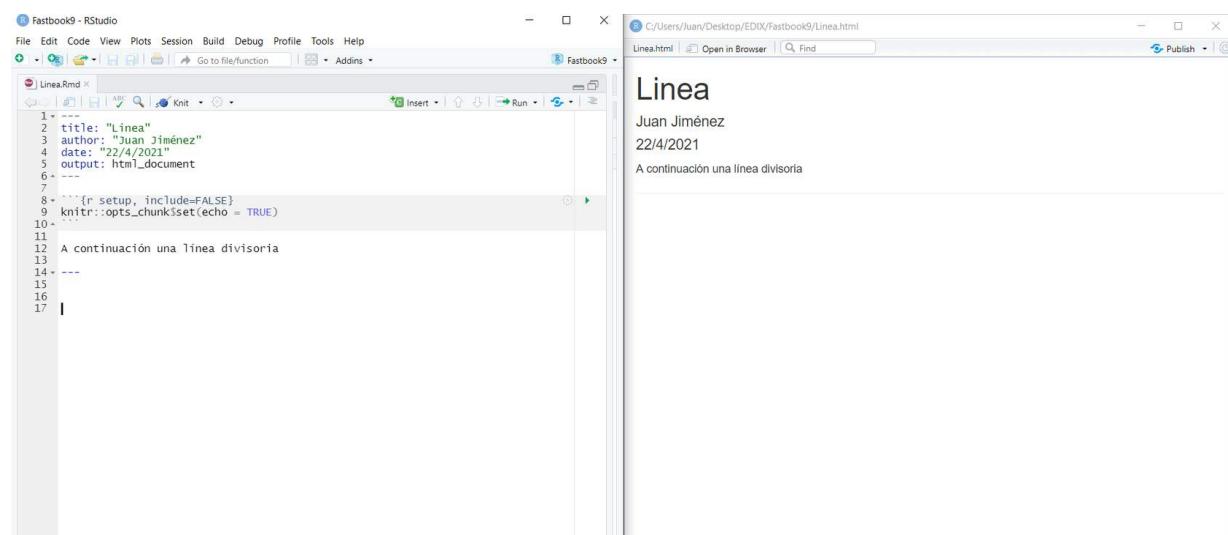
Markdown	Visualización
<pre> * Primer elemento * Segundo elemento * Tercer elemento * Cuarto elemento   * Primer elemento sublistা   * Segundo elemento sublistা  - Primer elemento - Segundo elemento - Tercer elemento  + Décimo elemento + Undécimo elemento + Duodécimo elemento </pre>	<ul style="list-style-type: none"> <li>• Primer elemento</li> <li>• Segundo elemento</li> <li>• Tercer elemento</li> <li>• Cuarto elemento           <ul style="list-style-type: none"> <li>◦ Primer elemento sublistা</li> <li>◦ Segundo elemento sublistা</li> </ul> </li>   <li>• Primer elemento</li> <li>• Segundo elemento</li> <li>• Tercer elemento</li>   <li>• Primer elemento</li> <li>• Segundo elemento</li> <li>• Tercer elemento</li> </ul>

Como ya hemos adelantado, en este segundo bloque vamos a ver algunos elementos más avanzados y especiales que merece la pena conocer.

## Línea horizontal

Markdown permite incluir líneas horizontales para dividir nuestro HMTL. Su aspecto es bastante elegante. Para introducirlas podemos usar **tres asteriscos**, **tres guiones** o **tres guiones bajos** (a nuestra elección).

La siguiente imagen muestra a la izquierda el documento markdown y a la derecha su renderización (el resultado).



The screenshot shows the RStudio interface with two panes. The left pane displays the Markdown file 'Linea.Rmd' with the following content:

```
1 ---  
2 title: "Linea"  
3 author: "Juan Jiménez"  
4 date: "22/4/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10  
11 A continuación una linea divisoria  
12  
13 ---  
14  
15  
16  
17 |
```

The right pane shows the rendered HTML output titled 'Linea'. The page header includes the title 'Linea', author 'Juan Jiménez', and date '22/4/2021'. Below the header is the text 'A continuación una linea divisoria' followed by a horizontal line.

## Citas

Si nos interesa introducir citas o resaltar ciertos mensajes de nuestro documento, podemos usar los denominados *blockquotes*. Para ello debemos usar el símbolo mayor que (>).

La siguiente imagen muestra a la izquierda el documento markdown y a la derecha su renderización (el resultado).

The screenshot shows the RStudio environment. On the left, the 'Fastbook9.Rmd' file contains the following R Markdown code:

```

1+ ---
2+ title: "Citas"
3+ author: "Juan Jiménez"
4+ date: "22/4/2021"
5+ output: html_document
6+ ---
7+
8+ > Este es el formato en el que se presentan las citas de un autor
9+ knitr::opts_chunk$set(echo = TRUE)
10+
11> Este es el formato en el que se presentan las citas de un autor
12+
13+
14|

```

On the right, the browser window displays the rendered HTML page titled 'Citas' with the author 'Juan Jiménez' and the date '22/4/2021'. The content of the page is: 'Este es el formato en el que se presentan las citas de un autor'.

## Código

Markdown también nos ayuda en la introducción de bloques de código. Si aplicamos **dos tabuladores o cuatro espacios** en el texto que escribimos, se detectará como código y se presentará como tal.

La siguiente imagen os muestra un ejemplo.

The screenshot shows the RStudio environment. On the left, the 'Codigo.Rmd' file contains the following R Markdown code:

```

1+ ---
2+ title: "Código"
3+ author: "Juan Jiménez"
4+ date: "22/4/2021"
5+ output: html_document
6+ ---
7+
8+ > #Si usamos dos tabuladores el texto aparece como código
9+ if(resultado>0){
10+   print(resultado)
11+
12+ }
13+
14+
15+
16+
17|

```

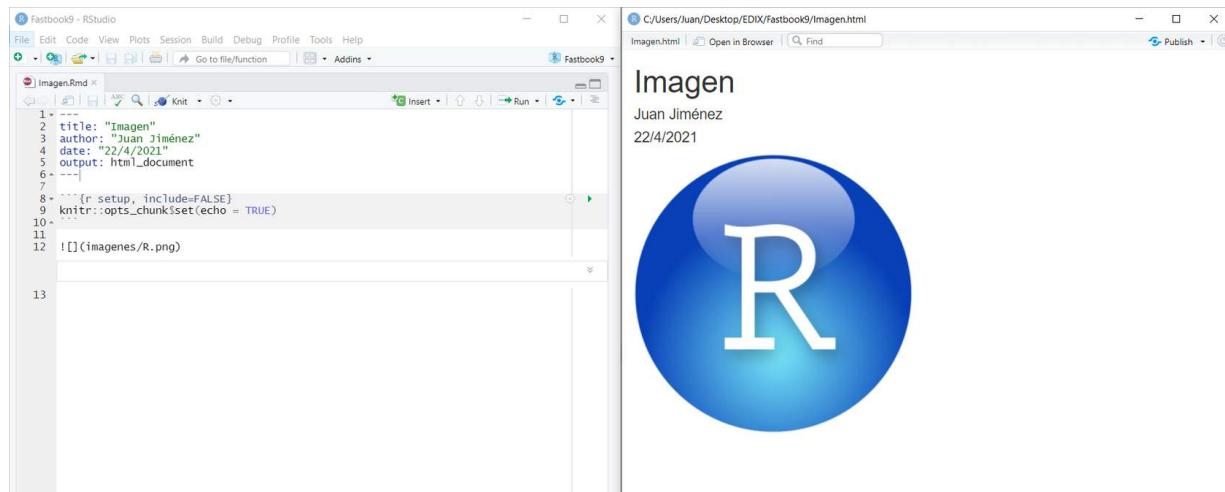
On the right, the browser window displays the rendered HTML page titled 'Código' with the author 'Juan Jiménez' and the date '22/4/2021'. The content of the page is: '#Si usamos dos tabuladores el texto aparece como código' followed by a block of code: 'if(resultado>0){ print(resultado) }'.

## Imágenes

Como no podía ser de otro modo, también tenemos la opción de incluir imágenes. Para ello, debemos usar la siguiente sintaxis (el texto en cursiva es aquel que habría que modificar).

*![Nombre\_alternativo\_en\_caso\_de\_que\_no\_cargue\_la\_imagen](url\_a\_imagen)*

Con el siguiente ejemplo lo podemos entender de una forma mucho más clara.



## Links y URLs

Por último, si necesitamos introducir URLs o links, podemos usar las siguientes dos sintaxis:

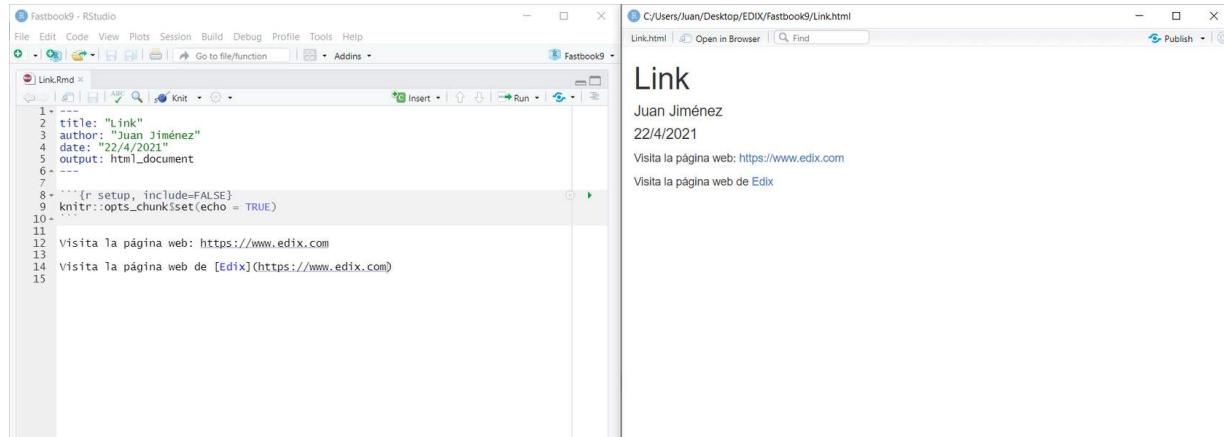
1

Para que se muestre la URL: <https://www.google.com>

2

Para que se muestre un texto diferente al de la URL: [Mi buscador favorito] (<https://www.google.com>)

En el siguiente ejemplo se encuentran aplicadas ambas casuísticas.



The screenshot shows two windows from RStudio. The left window is 'Link.Rmd' with the following R code:

```
1 title: "tlink"
2 author: "Juan Jiménez"
3 date: "22/4/2021"
4 output: html_document
5 ---
6
7 ````{r setup, include=FALSE}
8 knitr::opts_chunk$set(echo = TRUE)
9
10
11
12 Visita la página web: https://www.edix.com
13
14 Visita la página web de [Edix](https://www.edix.com)
15
```

The right window is 'Link.html' showing the generated HTML output:

**Link**  
Juan Jiménez  
22/4/2021  
Visita la página web: <https://www.edix.com>  
Visita la página web de [Edix](#)

# Conclusiones

**X** Edix Educación

---

Este fastbook nos ha servido como introducción y primer contacto a los documentos R Markdown. Creedme cuando os digo que es **una de las herramientas más útiles y valiosas que existen en el mundo de R**. En el futuro podréis echar mano de él para presentar vuestros resultados, análisis y modelos.

En el siguiente tema continuaremos trabajando sobre este mismo elemento desde una perspectiva más avanzada. Aprenderemos a introducir tablas, gráficos, índice de contenido y maquetación CSS, lo que dará a nuestros reportes un acabado mucho más profesional.

¡Enhорabuena! Fastbook superado

edix

Creamos Digital Workers