

Quarto Markdown Básico: Parte 1

Pedro Luis Luque

09/01/2025

Tabla de contenidos

1	Cabecera de nivel 1	1
1.1	Cabecera de nivel 2	2
1.1.1	Cabecera de nivel 3	2
1.1.1.1	Cabecera de nivel 4	2
	Cabecera de nivel 3 sin numerar	4
2	Más elementos markdown	4
2.1	Listado sin numerar	4
2.2	Listado numerado	4
2.3	Listados mixtos	5
2.4	Términos clave y matemáticas en línea	5
2.5	Uso del tanto por mil en Demografía	6
3	Escribir Código	8
3.1	Ejemplo con R	8
3.2	Ejemplo con Python	9
3.3	Uso de opciones en el código (“chunks”)	9
3.3.1	Opciones para código	9
3.4	Otras opciones para el código	10

1 Cabecera de nivel 1

Es fácil crear texto con Markdown: *itálicas*, **negrita**, y ***negrita itálica***, al escribir: `*itálicas*`, `**negrita**`, y `***negrita itálica***`.

1.1 Cabecera de nivel 2

Se pueden escribir superíndices y subíndices para exponentes como 3^2 o H_2O , al escribir: `3^2` o `H~2~O`. Observa el uso de dos símbolos para envolver el elemento que se está moviendo hacia arriba o hacia abajo.

La cabecera de nivel 2 se formatea con un estilo especial tiene automáticamente una línea horizontal debajo (en html).

1.1.1 Cabecera de nivel 3

También se puede utilizar texto tachado ~~incorrecto~~, al escribir `~~incorrecto~~`.

1.1.1.1 Cabecera de nivel 4

Se pueden usar “acentos descendentes”, “acentos graves” o “Backticks” para indicar **texto literal**, al escribir: ``texto literal``. Eso puede ser útil para escribir código ya que usa una fuente monoespacial (todas las letras ocupan el mismo espacio).

1.1.1.1.1 Cabecera de nivel 5

Enlazar a otras páginas web es fácil. Si se quiere mostrar un enlace, envolver el enlace entre `<` y `>` como en el siguiente ejemplo: <http://destio.us.es/calvo>, al escribir: `<http://destio.us.es/calvo>`.

También puede usarse corchetes y paréntesis para indicar un enlace en línea como el siguiente: [Mi página personal](#), al escribir: `[Mi página personal](http://destio.us.es/calvo)`.

Cabecera de nivel 6

Es fácil agregar **código** a su documento Quarto utilizando una pareja de tres “acentos graves” o “backticks”: `````. Puede definir el lenguaje dentro de las llaves y luego usar el estilo de ese lenguaje para su formateo. En el siguiente código R se suma el valor de dos variables e imprime el resultado.

```
# <-- Este símbolo se usar para comenzar una línea de comentario al escribir R
# las líneas comentadas no se ejecutarán como parte del código
# se pueden dejar comentarios útiles para ti u otras personas que lo lean

# crear una variable y asignarle su valor
variable_1 = 3

#crear una segunda variable y darle su valor
```

```
variable_2 = 3

#añadir las dos variables y grabar el resultado como una tercera variable
variable_3 = variable_1 + variable_2

#print muestra el resultado guardado en la variable
print(variable_3)
```

[1] 6

Al escribir:

```
# <-- Este símbolo se usar para comenzar una línea de comentario al escribir R
# las líneas comentadas no se ejecutarán como parte del código
# se pueden dejar comentarios útiles para ti u otras personas que lo lean

# crear una variable y asignarle su valor
variable_1 = 3

#crear una segunda variable y darle su valor
variable_2 = 3

#añadir las dos variables y grabar el resultado como una tercera variable
variable_3 = variable_1 + variable_2

#print muestra el resultado guardado en la variable
print(variable_3)
```

[1] 6

También se puede **incluir información guardada en objetos R en la redacción de un párrafo**. Por ejemplo, el valor de `variable_3` es igual a 6.

Para mostrarlo el valor de un objeto R se tiene que escribir: un acento grave + “r”, seguido del objeto R (o cálculo en R), y finalizar con un acento grave. Para obtener el párrafo anterior se ha escrito en el fichero Quarto:

```
También se puede **incluir información guardada en objetos R en la redacción de un párrafo**. Por ejemplo, el valor de `variable_3` es igual a `r variable_3`.
```

Cabecera de nivel 3 sin numerar ni en la tabla de contenido

Texto ...

Cabecera de nivel 3 sin numerar

Texto ...

2 Más elementos markdown

A continuación se muestran algunos ejemplos de markdown básicos. El código markdown que se escribe para obtener estos resultados puede verse en el fichero fuente Quarto (qmd) que se ha utilizado para crear este documento.

2.1 Listado sin numerar

Estos listados se pueden escribir poniendo al inicio de cada elemento un símbolo: *, - o +.

- Elemento A
- Elemento B
- Elemento C
- Elemento D
- Elemento E

2.2 Listado numerado

Estos listados se pueden escribir poniendo al inicio de cada elemento:: 1. (número+“.”), a., i. o A.

1. Elemento A
2. Elemento B
3. Elemento C
4. Elemento D
5. Elemento E

2.3 Listados mixtos

1. Elemento A
 - Elemento A.1
 - Elemento A.2
 - a. Elemento A.2.a
 - b. Elemento A.2.b
 - c. Elemento A.2.c
2. Elemento B
3. Elemento C

Para este tipo de listados anidados se usan “tabulaciones de 4 espacios”. Este listado se ha escrito con el siguiente código:

```
1. Elemento A
  * Elemento A.1
  * Elemento A.2
    a. Elemento A.2.a
    b. Elemento A.2.b
    c. Elemento A.2.c
2. Elemento B
3. Elemento C
```

2.4 Términos clave y matemáticas en línea

Primer término definición del primer término.

Segundo término definición del segundo término.

Tercer término definición del tercer término.

Fórmula $i^2 = j^2 = k^2 = ijk = -1$, que se escribe: $i^2 = j^2 = k^2 = ijk = -1$.

Se puede usar matemáticas resaltadas (centradas) al escribir una pareja de dos símbolos del dólar: \$\$.

$$i^2 = j^2 = k^2 = ijk = -1$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$\sum_{i=1}^n (\bar{x} - x_i)^2$$

Las anteriores ecuaciones, se han obtenido al escribir el código LaTeX siguiente:

```
$$
i^2 = j^2 = k^2 = ijk = -1
$$
```

```
$$
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
$$
```

```
$$
\sum_{i = 1}^n {(\bar{x} - x_i)^2}
$$
```

La Ecuación 1 muestra el Teorema de Bayes y la Ecuación 2 muestra una fórmula usada en regresión lineal (se ha incluido referencias a fórmulas):

$$Pr(\theta|y) = \frac{Pr(y|\theta)Pr(\theta)}{Pr(y)} \quad (1)$$

$$Y \sim \beta_0 + \beta_1 X + \epsilon \quad (2)$$

Se han obtenido al escribir:

```
$$
Pr(\theta | y) = \frac{Pr(y | \theta) Pr(\theta)}{Pr(y)}
$$ {\#eq-bayes}
```

```
$$
Y \sim \beta_0 + \beta_1 X + \epsilon
$$ {\#eq-regresion}
```

2.5 Uso del tanto por mil en Demografía

- Para usarlo en salidas “html” se escribe: `‰` (‰)
- Para usarlo en salidas “pdf” se puede usar:
 - El mismo código que en html (método recomendado): `‰` (‰)

- El símbolo del tanto por mil en LaTeX es: `\permil`, se escribe en markdown: `\permil` y produce: ‰. Pero para que pueda verse correctamente, se necesita añadir en la cabecera yaml del documento:

```
header-includes:  
- \usepackage{wasysym}
```

Con esta opción no se vería en html, de ahí que se recomienda usar el primer método.

i Salto de página en pdf

Para que se produzca un salto de página en un fichero de salida “pdf”, se escribe el siguiente código markdown:

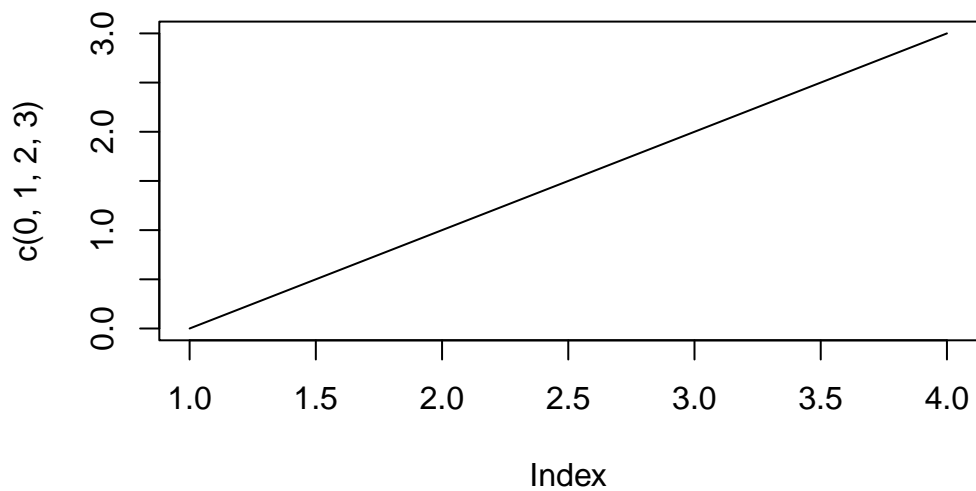
```
{{< pagebreak >}}
```

3 Escribir Código

3.1 Ejemplo con R

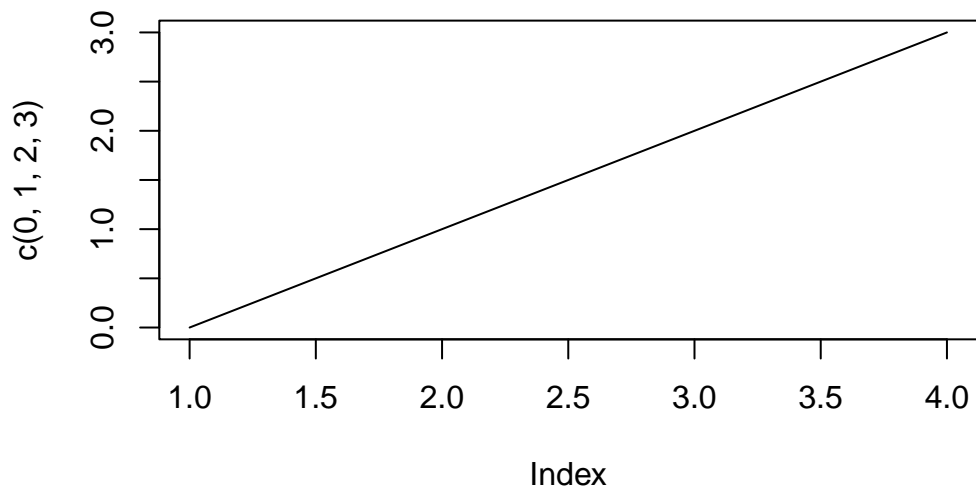
Un ejemplo de gráfico lineal usando el sistema base de R “graphics”.

```
plot(c(0,1,2,3), type="l")
```



Se obtiene al escribir:

```
plot(c(0,1,2,3), type="l")
```



3.2 Ejemplo con Python

Un ejemplo de gráfico linear usando Python y la librería `matplotlib`.

```
import matplotlib.pyplot as plt
plt.plot([0,1,2,3])
plt.show()
```

En este caso, para que no se evalúe el código y de ahí que no se muestre el gráfico resultante (solamente se ve el código), se ha añadido la opción de chunk: `eval: false`. El código escrito ha sido el siguiente:

```
import matplotlib.pyplot as plt
plt.plot([0,1,2,3])
plt.show()
```

3.3 Uso de opciones en el código (“chunks”)

```
#crea un data.frame desde dos vectores e imprime el data.frame
equipo    = c("A", "B", "C")
ganancia  = c(45,63,47)
df        = data.frame(equipo, ganancia)
print(df)
```

	equipo	ganancia
1	A	45
2	B	63
3	C	47

3.3.1 Opciones para código

Hay dos opciones importantes:

- `echo`: incluye el código fuente en el documento de salida
- `warning`: incluye “avisos” o “warnings” en el documento de salida

Examine el siguiente código escrito en el fichero Quarto:

```
# crea dos variables con valores, almacena la suma en una tercera variable
# e imprime la tercera variable
v1 = 5
v2 = 4
v3 = v1 + v2
print(v3)
```

[1] 9

La salida que se muestra en el fichero (“html”, “pdf”, ...) es la siguiente:

```
# crea dos variables con valores, almacena la suma en una tercera variable
# e imprime la tercera variable
v1 = 5
v2 = 4
v3 = v1 + v2
print(v3)
```

[1] 9

En la mayoría de los informes, se quiere que únicamente se muestren los resultados y no se muestre el código utilizado en ninguna parte. También se podría mostrar puntualmente algún código porque se quiera resaltar.

- `#| echo: false`
- `#| warning: false`

[1] 9

La opción `echo: false` desactiva la impresión del código (solamente muestra los resultados).

3.4 Otras opciones para el código

También se puede utilizar en la cabecera del documento yaml estas opciones y otras de forma general

```

---
title: "Escribir código R en un documento Quarto"
author: "Pedro Luis Luque"
date: "`r Sys.Date()`" # "1-16-2024" # daría error: 16-1-2024
date-format: "DD/MM/YYYY"
lang: es
toc: true
toc-depth: 4
number-sections: true
format:
  html:
    code-tools: true
    code-fold: true
  pdf: default
execute:
  echo: true
  warnings: false
---

```

- ① `title:`, se escribe el título del documento.
- ② `lang: es`, se está indicando que el texto está escrito en español.
- ③ `toc: true`, con esta opción se construye una tabla de contenido o índice del documento.
- ④ `toc-depth: 4`, indica que en el índice se muestren hasta los títulos de nivel 4 (####).
- ⑤ `number-sections: true`, indica que se quiere enumerar las títulos o secciones del documento.
- ⑥ `format:`, se inicia el apartado de las salidas que se quieren generar.
- ⑦ `html:`, se genera la salida de ficheros “html”.
- ⑧ `code-tools: true`, activa la consulta del código markdown-quarto utilizado para generar el documento (para ficheros “html”).
- ⑨ `code-fold: true`, permite plegar u ocultar (“folding”) el código o mostrarlo desplegado.
- ⑩ `pdf:`, se genera la salida de ficheros “pdf”.
- ⑪ `execute:`, se inicia el apartado de cómo ejecutar el código incluido en el fichero Quarto.
- ⑫ `echo: true`, indica que por defecto el código que se escriba en un chunk se mostrará (con `echo: false`, no se mostraría el código).
- ⑬ `warnings: false`, indica que los mensajes de aviso o warnings que se produzcan no aparezcan en la salida.

Esta es una representación de las muchísimas opciones que se pueden establecer en la **cabecera yaml de un documento Quarto**.

Nota: más información sobre uso de fechas (“`date:`”) en la cabecera yaml: [Documentación Quarto](#).