Elementos de Rmarkdown

David García Callejas

ELEMENTOS DE UN ARCHIVO Rmd

- BLOQUES DE CÓDIGO (chunks)
  - mostrar/ocultar código fuente: `echo`
  - mostrar/ocultar output: `results`
  - mostrar/ocultar warnings: `warning`
  - mostrar/ocultar mensajes: `message`
  - ocultar figuras: `fig.show="hide"`
  - mostrar/ocultar todo lo anterior (pero ejecutando el bloque): `include`
  - unir outputs en un sólo bloque: `collapse`
  - evaluar el código del bloque: `eval`
  - opciones globales: `knitr::opts_chunk\$set(...)\`
- FORMATO DE TEXTO
  - SECCIONES:
    - NO NUMERADAS: `#` antes del nombre de la sección

    ```
    # Sección
    ## Sub-sección
    ### etc
    #### etc
    ```

    - NUMERADAS:

    Incluyendo `number_sections` en el encabezado:

    ```
    output:
      html_document:
        number_sections: true
      pdf_document:
        number_sections: true
    ```

    Se puede añadir una sección no numerada dentro de un documento con secciones numeradas:

    ```
    # Sección sin numerar {-}
    ```

  - CURSIVA:

    `*texto en cursiva*`, o `_texto en cursiva_`

  - NEGRITA:

    `**texto resaltado**`, o `__texto resaltado__`

  - SUBRAYADO:

    No existe por defecto en markdown; se puede hacer indirectamente usando LaTeX:

    ```
    $\text{\underline{Texto subrayado usando sintaxis \LaTeX}}$
    ```

- TAMAÑO DE LA FUENTE:

  Incluir en el encabezado:

  ```
  fontsize:12pt
  ```

- COLOR DE LA FUENTE:

  Podemos usar sintaxis LaTeX:

  ```
  $\color{red}{\text{texto en rojo}}$
  ```

  texto en rojo

  Si generamos output en html, también podemos usar sintaxis html:

  ```
  <span style="color: red;">texto en rojo para html</span>
  ```

- TIPO DE FUENTE:

  Esta opción no se puede incluir directamente en el archivo `.Rmd`. Necesitamos crear un archivo auxiliar que nos indique el "estilo" que usará el traductor LaTeX para generar el documento final. En la misma carpeta donde tenemos el archivo `.Rmd`, creamos un archivo con extensión `.sty`. Si queremos usar la fuente "Biolinum sans serif", éste sería el contenido de nuestro archivo `mi_estilo.sty`:

  ```
  \usepackage{biolinum}
  \renewcommand{\familydefault}{\sfdefault} % sans serif
  \fontfamily{ppl}\selectfont
  ```

  Fijáos que, además de seleccionar el nombre de la fuente (biolinum), indica que debe usar la fuente sans serif por defecto, (la línea "renewcommand"). Si escogéis una fuente serif, esta línea no es necesaria. El catálogo de fuentes disponibles en LaTeX se puede consultar en https://tug.org/FontCatalogue/.

  Para que estos cambios tengan efecto en un archivo `.Rmd`, tenemos que referenciar el estilo en el encabezado:

  ```
  ---
  output:
  pdf_document:
      includes:
          in_header: mi_estilo.sty
  ---
  ```

- ENLACES WEB:

  ```
  [texto en el documento](dirección web)
  ```

- LISTAS NUMERADAS:

  ```
  1. Primer elemento
  2. Segundo elemento
     1. sub-elemento

        texto en sub-elemento 1
     2. sub-elemento 2
  ```

  Es importante el punto al final de cada número, empezar cada sublista a la altura del primer carácter del nivel superior, y dejar una línea en blanco entre el encabezado del nivel y el texto.

- LISTAS NO NUMERADAS:

```
* Primer elemento
* Segundo elemento
  + sub-elemento
    - sub-sub
  + sub-elemento 2
```

- PÁRRAFOS/SALTOS DE LÍNEA:

  Para empezar un nuevo párrafo, añade dos espacios al final del anterior o una línea de separación:

  Una frase que debería ser final de párrafo. ¿Otro párrafo?

  Una frase que debaría ser final de párrafo.
  Al añadir dos espacios en la frase anterior (o una línea entre ambas), funciona.

  La opción `<br>` también crea un salto de línea.

- LINEAS HORIZONTALES:

  Tres asteriscos o tres guiones (separados del resto del texto por saltos de línea) generan una línea horizontal: `***`/`---`

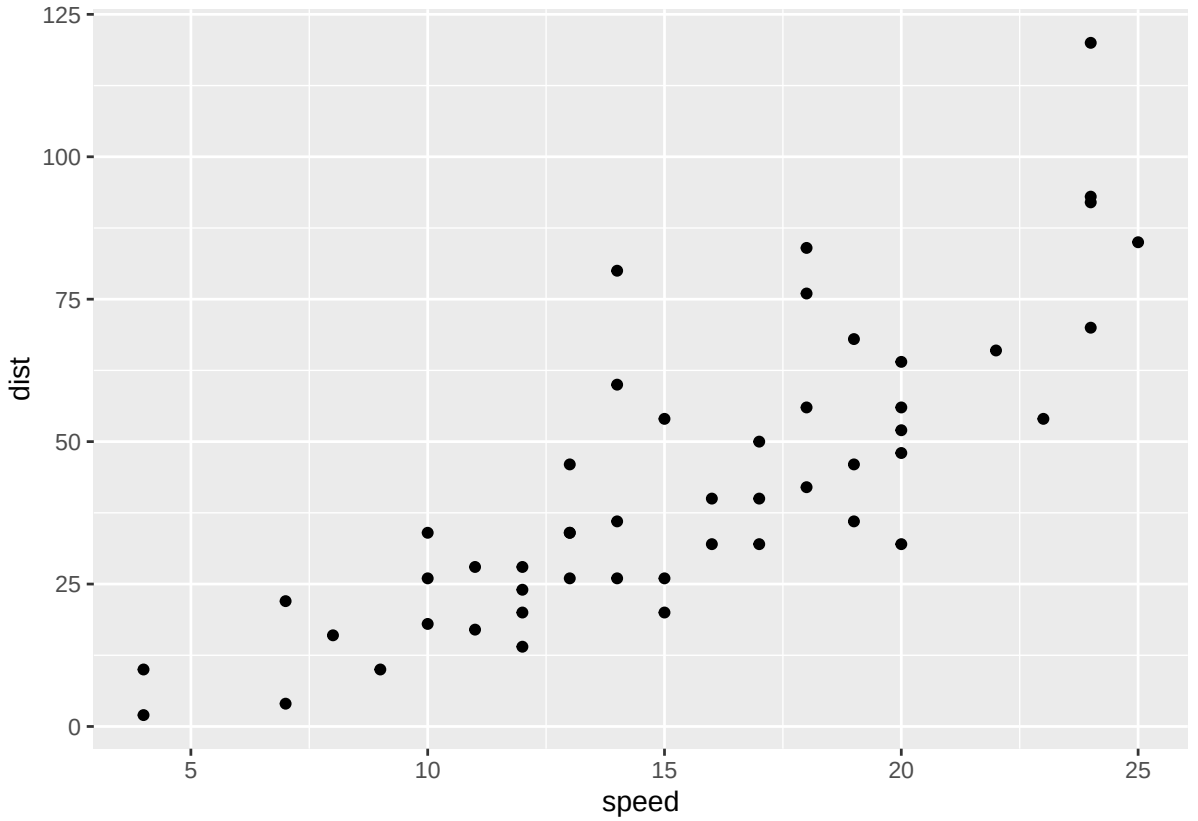
- SALTOS DE PÁGINA:

  `\newpage`

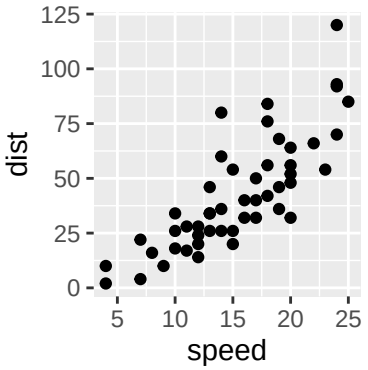
- FIGURAS DINÁMICAS Y EXTERNAS

Rmarkdown permite incluir en el documento generado tanto imágenes dinámicas, generadas dentro de R, como externas. Para el primer caso, basta con generar la figura dentro de un bloque de código de R.

```
ggplot(cars, aes(speed, dist)) + geom_point()
```

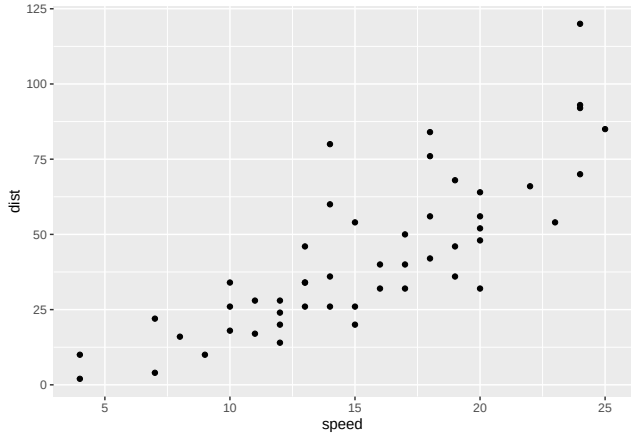


Se puede ajustar el tamaño y la resolución de las imágenes generadas, a través de las opciones `fig.width` y `fig.height`, que tienen unidades de pulgadas. Por ejemplo, la imagen a continuación se genera con los parámetros `{r fig.width = 2,fig.height = 2}`:



Una alternativa más intuitiva puede ser usar las opciones `out.width` y `out.height`, que aceptan valores de porcentaje con respecto al marco donde se encaja la imagen:

````
```{r out.width = "50%"}
ggplot(cars, aes(speed, dist)) + geom_point()
```
````



Hay dos maneras de incluir imágenes externas. La primera, muy sencilla, tiene la sintaxis `![texto pie de imagen](ruta_imagen)`. Por ejemplo, la figura a continuación se genera con la orden `![Lince ibérico](../data/figuras/Lynx_pardinus.jpg)`. La ruta a la imagen se entiende *relativa* a la localización del archivo `.Rmd`.

Figura 1: Lince ibérico

Esta sintaxis no permite mucha flexibilidad. Para modificar el tamaño u otros aspectos de la imagen, es más apropiado importar imágenes dentro de bloques de código, con la función `include_graphics`. En la definición del bloque podemos definir opciones como el alineado (`fig.align`), o el tamaño, como vimos arriba (`out.width` y `out.height`). También podemos especificar el pie de figura con la opción `fig.cap`. Para ello, nos aseguraremos de que en el encabezado del documento tengamos la siguiente opción:

```
  pdf_document:
    fig_caption: true
```

El código y la figura con las opciones elegidas:

````
```{r lince, fig.align = 'center', out.width = "50%", fig.cap='Lince ibérico, segunda versión'}
    knitr::include_graphics(here::here("data/figuras", "Lynx_pardinus.jpg"))
```
````

Figura 2: Lince ibérico, segunda versión

- TABLAS

  Rmarkdown permite generar tablas "manualmente":

variable	valor
v1	el valor de v1.
v2	el valor de v2.

  Existen multitud de opciones para personalizar tablas de este estilo, pero no las veremos aquí (se pueden consultar en el manual de Pandoc). En general, es mucho más interesante generar tablas de manera automática a partir de datos que ya tengamos en matrices o dataframes. Para ello, una de las opciones más utilizadas es a través de la función `kable` de `knitr`:

```
knitr::kable(mtcars[1:5,])
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

Cuadro 2: tabla básica con kable

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.88	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.21	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2

Al igual que con las imágenes, podemos añadir un pie de tabla, entre otras opciones

```
knitr::kable(mtcars[1:5,],
             caption = "tabla básica con kable",
             digits = 2)
```

Además de `kable`, existen varios paquetes que también permiten generar tablas en varios formatos, como el paquete `stargazer`:

````
```{r results='asis', echo = FALSE}
stargazer::stargazer(mtcars[1:5,],
summary = FALSE,
header = FALSE,
title = "tabla básica con stargazer")
```
````

Cuadro 3: tabla básica con stargazer

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.900	2.620	16.460	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.900	2.875	17.020	0	1	4	4
Datsun 710	22.800	4	108	93	3.850	2.320	18.610	1	1	4	1
Hornet 4 Drive	21.400	6	258	110	3.080	3.215	19.440	1	0	3	1
Hornet Sportabout	18.700	8	360	175	3.150	3.440	17.020	0	0	3	2

Es importante remarcar que `stargazer` lo que hace es generar código LaTeX o html que será traducido al documento final. Al generarse estas tablas dentro de bloques de código, `knitr` tiene que entender que el output de un bloque así no es el resultado de ninguna operación en R, sino que se debe compilar como si fuera texto. Esto se especifica con la opción `results='asis'` en la definición de los bloques en los que tengamos tablas `stargazer`.

Y también, de igual forma que sucede con las imágenes, es posible que las tablas no aparezcan exactamente en el lugar del documento donde las hemos definido (esto es un problema recurrente en LaTeX). Para tener mayor control sobre el posicionamiento de nuestras tablas, podemos usar el paquete `kableExtra` con la siguiente opción:

```
knitr::kable(mtcars[1:5,],
             caption = "The first 5 rows of the mtcars data",
              digits = 2) %>%
kableExtra::kable_styling(latex_options = "hold_position")
```

Esta solución, sin embargo, puede generar cambios en el estilo de las tablas, por la diferente configuración de `kable` y `kableExtra` (https://github.com/haozhu233/kableExtra/issues/265).

Una solución más general para solucionar el posicionamiento de figuras y tablas (al generar pdfs) es forzar al traductor de LaTeX a mantener el posicionamiento original. Para ello deberíamos incluir código de LaTeX en

nuestro documento .Rmd. Esto se puede hacer incluyendo el archivo .tex que queramos referenciar, en el encabezado del documento .Rmd. En este caso usamos el siguiente código de LaTeX:

```
\usepackage{float}
\let\origfigure\figure
\let\endorigfigure\endfigure
\renewenvironment{figure}[1][2] {
  \expandafter\origfigure\expandafter[H]
} {
  \endorigfigure
}
```

Estas líneas las guardamos en un archivo .tex en el mismo directorio que nuestro archivo .Rmd, por ejemplo `latex_options.tex`. En el encabezado del documento .Rmd, incluiremos:

```
output:
pdf_document:
  fig_caption: yes
  includes:
    in_header: latex_options.tex
```

Por último, es muy habitual que queramos mostrar en una tabla el resultado de análisis estadísticos. Para ello, la mejor opción es usar -de nuevo- el paquete `stargazer`:

```
ejemplo <- lm(Petal.Width ~ Petal.Length,data = iris)

stargazer::stargazer(ejemplo,
header = FALSE,
label = "tab:regresion1",
type=ifelse(knitr::is_latex_output(),"latex","html"),
title="tabla de regresión con stargazer")
```

Cuadro 4: tabla de regresión con stargazer

	Dependent variable:
	Petal.Width
Petal.Length	0.416^{***}
	(0.010)
Constant	-0.363^{***}
	(0.040)
Observations	150
R^2	0.927
Adjusted R^2	0.927
Residual Std. Error	0.206 (df = 148)
F Statistic	$1{,}882.452^{***}$ (df = 1; 148)
Note:	*p<0.1; **p<0.05; ***p<0.01

- PIES DE FIGURAS Y TABLAS

  Por defecto, los pies de figura son creados con la forma `Figure X:`, y los pies de tabla, con `Table X:`. Es posible elegir el idioma en el que se generan, incluyendo la opción `lang: es` (en el caso del español) en el encabezado como línea independiente. Igualmente, se pueden personalizar incluyendo el siguiente código directamente en el .Rmd o en un archivo .tex referenciado:

```
\def\figurename{MiPieDeFigura}
\def\tablename{MiPieDeTabla}
```

- EXPRESIONES MATEMÁTICAS

  Rmarkdown puede aprovechar la potente sintaxis de LaTeX para generar todo tipo de expresiones matemáticas. Se pueden incluir expresiones matemáticas o bien entre el texto, encapsulándolas entre dos símbolos de dólar `$ -expresión- $`, por ejemplo para mostrar pequeñas expresiones ($\lambda = f - 1$), o bien como expresiones independientes, con la sintaxis:

```
\begin{equation}
  mi-ecuación
\end{equation}
```

Por ejemplo:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{1}$$

La sintaxis para expresiones matemáticas de LaTeX es enormemente rica. El ejemplo anterior muestra fracciones, sumatorios, subíndices, etc. Se escribiría así: `\bar{X} = \frac{\sum_{i=1}^n X_i}{n}`. Cada elemento se estructura con una barra invertida, el nombre del elemento que queremos, y entre llaves, la expresión en sí. Por ejemplo, para una fracción, usamos la orden `\frac{numerador}{denominador}`.

En este enlace hay una lista de todos los símbolos utilizables: https://oeis.org/wiki/List_of_LaTeX_mathematical_symbols; en este otro enlace, una explicación detallada de esta sintaxis: https://en.wikibooks.org/wiki/LaTeX/Mathematics.

- REFERENCIAS
  - REFERENCIAS A FIGURAS, TABLAS, ECUACIONES

    Para usar esta funcionalidad, debemos incluir en el encabezado la siguiente opción (sustituyendo 'pdf' por 'html' o 'word', si queremos otro formato):

    ```
    output:
      bookdown::pdf_document2: default
    ```

    Si tenemos otras opciones en el encabezado, como incluir las opciones de LaTeX que vimos para posicionar figuras o tablas, y queremos generar tanto pdf como html, el encabezado completo quedaría similar a este:

    ```
    output:
      bookdown::pdf_document2:
        fig_caption: yes
        includes:
          in_header: latex_options.tex
      bookdown::html_document2:
        fig_caption: yes
        df_print: paged
    ```

    Cada figura o tabla debe estar incluida en un único bloque con su id correspondiente, y además, debe tener un pie de figura (caption). Por ejemplo, el bloque que vimos más arriba para mostrar una figura externa tiene la id 'lince':

    ````
    ```{r lince, fig.align = 'center', out.width = "50%",
        fig.cap='Lince ibérico, segunda versión'}
    knitr::include_graphics(here::here("data/figuras", "Lynx_pardinus.jpg"))
    ```
    ````

Podemos referenciar esta figura con la orden `\@ref(fig:lince)`. En el documento final quedará como una referencia a la figura 2. De la misma manera, podemos referenciar tablas (de nuevo, una tabla dentro de un único bloque de código): `\@ref(tab:'id de bloque')`. Por ejemplo, con el bloque llamado `mitabla2`, más arriba: `\@ref(tab:mitabla2)` genera una referencia a la tabla 2. Fijaos en la sintaxis de stargazer (tabla 4), que es ligeramente diferente.

También se pueden referenciar ecuaciones. Para ello, al escribir la ecuación (como vimos más arriba), incluimos un id que nos permita referenciarla más adelante:

```
\begin{equation}
  \bar{X} = \frac{\sum_{i=1}^n X_i}{n} \label{eq:media}
\end{equation}
```

La parte `(\#eq:media)` es la que me permite referenciarla. Para ello, al igual que con figuras o tablas, escribo `\@ref(eq:media)`, (aquí la referencia a la ecuación (1)).

- REFERENCIAS A PUBLICACIONES/DOCUMENTOS EXTERNOS

Es posible generar referencias dinámicas a documentos externos e incluir una sección de 'bibliografía' al final de nuestros archivos .Rmd. Para ello, necesitamos varios elementos. El primero de ellos es una lista de los documentos que queremos, potencialmente, referenciar. Esta lista estará descrita en un archivo externo con extensión '.bib', que almacenaremos en la misma carpeta que nuestro archivo .Rmd. Los archivos .bib contienen una entrada por cada referencia disponible, y cada entrada tiene una sintaxis de este tipo:

```
@Book{ggplot2,
  author = {Hadley Wickham},
  title = {ggplot2: Elegant Graphics for Data Analysis},
  publisher = {Springer-Verlag New York},
  year = {2016},
  isbn = {978-3-319-24277-4},
  url = {http://ggplot2.org},
}
```

Después de la primera '@' especificamos el tipo de documento (Book, en el ejemplo). Dentro de las llaves, la primera palabra es el 'id' del documento, el nombre por el que lo referenciaremos en nuestro .Rmd. Tras este identificador, vienen una serie de campos como el autor, título, etc.

En la mayoría de los casos, no será necesario escribir estos archivos .bib a mano. Cualquier gestor de referencias nos permite exportar toda una serie de referencias en este formato de manera automática (Zotero, Mendeley, EndNote, etc). Además, buscadores como Google Scholar también permiten exportar cualquier referencia en formato .bib (en Scholar, pinchar en las comillas debajo de un artículo, y seleccionar 'Bibtex').

Una vez tenemos todas nuestras referencias en un archivo, por ejemplo 'referencias.bib', el siguiente paso es incluirlo en el encabezado de nuestro documento .Rmd, con la opción

```
bibliography: referencias.bib
```

Cuando nuestro 'referencias.bib' esté incluido en el encabezado, ya podremos citar sus elementos. En este ejemplo, he incluido dos referencias, la primera con la id 'Broman2018', y la segunda con id 'Rodriguez-Sanchez2016'. Para citarlos, basta con escribir, por ejemplo, `@Broman2018`, o bien entre corchetes si queremos la referencia entre paréntesis: `[@Broman2018]`. A continuación cito a Broman y Woo (2018) y a (Rodriguez-Sanchez et al. 2016). También podemos citar sólo el año (`[-@Broman2018]`), o citar páginas concretas (`[@Broman2018 2-4]`). Al citar al menos un documento externo, se genera automáticamente una lista de referencias al final del documento, de tal manera que podemos nombrar esa sección como queramos (en este documento, la he llamado 'referencias', y la muestro en su propia página nueva). Para 'mover' la sección de referencias,

se puede usar la orden `<div id="refs"></div>`, y las referencias aparecerán en ese punto del documento. A pesar de ser código html, debería funcionar también al compilar en pdf.

También es posible modificar el estilo que tendrán nuestras referencias (ver ejemplos en https://www.zotero.org/styles). Cada estilo viene especificado por otro archivo externo, en este caso con extensión '.csl'. Por ejemplo, si queremos utilizar el estilo que usa la revista 'Nature' para sus referencias, podemos bajar el archivo 'nature.csl' a nuestra carpeta, y referenciarlo en el encabezado de nuestro .Rmd:

```
csl: nature.csl
```

- NOTAS AL PIE

  Se pueden crear con esta sintaxis: `texto^[nota al pie]`. Por ejemplo[1].

- FORMATO DE PÁGINA
  - ÍNDICE

    Si usamos `bookdown`, se puede incluir o no de manera automática un índice ('toc', table of contents) con secciones numeradas (number_sections) pasando a `false` o `true` las siguiente opciones en el encabezado:

    ```
    bookdown::pdf_document2:
      number_sections: false
      toc: false
    ```

    Esta opción sólo está disponible para generar documentos en formato pdf, por el momento.

  - NÚMEROS DE LÍNEA

    Para incluir números de línea en todo el texto (pero no en los bloques de código), incluir en el encabezado:

    ```
    header-includes:
      - \usepackage[left]{lineno}
      - \linenumbers
    ```

    Para incluir números de línea en los bloques de código (pero no en el texto), incluimos la siguiente opción en los bloques que queramos numerar:

    ```
    attr.source='.numberLines'
    ```

    Por ejemplo:

```
  if (TRUE) {
    x <- 1:10
    x + 1
  }
```

```
##  [1]  2  3  4  5  6  7  8  9 10 11
```

```
Esta numeración se reinicia en cada bloque:
```

```
  y <- seq(1:5)
  x <- sqrt(y)
```

  - MÁRGENES

    Incluir en el encabezado las opciones de la clase 'geometry' de LaTeX:

    ```
    geometry: margin=1in
    ```

[1] esto es una nota

- TIPO DE RESALTADO DE CÓDIGO:

  Incluir en el encabezado el tipo de resaltado que queremos (las opciones son "default", "tango", "pygments", "kate", "monochrome", "espresso", "zenburn", y "haddock", se pueden consultar en https://www.garrickadenbuie.com/blog/pandoc-syntax-highlighting-examples/):

  ```
  output:
    pdf_document:
      highlight: tango
  ```

- ENCABEZADO/PIE DE PÁGINA

  Para encabezados y pies de página sencillos (como el encabezado de este documento), podemos incluirlos directamente en nuestro .Rmd, de nuevo añadiendo opciones al encabezado:

  ```
  header-includes:
    - \usepackage{fancyhdr}
    - \pagestyle{fancy}
    - \fancyhead[CO,CE]{Texto encabezado}
    - \fancyfoot[CO,CE]{Texto pie de página}
  ```

  Otra opción interesante es eliminar los números de página que vienen incluidos por defecto al "tejer" el documento en pdf.

  ```
  header-includes:
    - \pagenumbering{gobble}
  ```

  En general, el texto entre llaves es texto que acepta sintaxis LaTeX, por lo que podemos incluir números de página (`\thepage`) u otros elementos. Si queremos algo más detallado, podemos crear un archivo .tex con el encabezado/pie de página que deseemos, y referenciarlo en nuestro Rmd:

  ```
  output:
    pdf_document:
      includes:
        before_body: encabezado.tex
        after_body: pie.tex
  ```

  Podemos consultar los elementos más comunes en esta página: https://bookdown.org/yihui/rmarkdown-cookbook/latex-header.html. Estas opciones son válidas para generar documentos pdf. Para generar html, dado que no esperamos páginas diferenciadas, tiene menos sentido especificar un encabezado o pie de página.

ENCABEZADO COMPLETO

Podemos resumir varias de las opciones que hemos ido desgranando y mostrar cómo quedaría un encabezado que integre varias de ellas. En particular, éste es el encabezado del archivo .Rmd asociado a estos apuntes:

```
title: "Elementos de Rmarkdown"
author: "David García Callejas"
lang: es
output:
  bookdown::pdf_document2:
    number_sections: false
    fig_caption: yes
    toc: false
    includes:
      in_header: latex_options.tex
  bookdown::html_document2:
    number_sections: false
    fig_caption: yes
```

```
    df_print: paged
bibliography: referencias.bib
header-includes:
- \usepackage{fancyhdr}
- \pagestyle{fancy}
- \fancyhead[CO,CE]{Elementos de Rmarkdown}
```

Aplicaciones avanzadas

- WEBS CON BLOGDOWN
- LIBROS

Referencias

Broman, Karl W., y Kara H. Woo. 2018. «Data Organization in Spreadsheets». *The American Statistician* 72 (1): 2-10. https://doi.org/10.1080/00031305.2017.1375989.

Rodriguez-Sanchez, Francisco, Antonio Jesús Pérez-Luque, Ignasi Bartomeus, y Sara Varela. 2016. «Ciencia reproducible: qué, por qué, cómo». *Revista Ecosistemas* 25 (2): 83-92-92. https://doi.org/10.7818/re.2014.25-2.00.