

Escritura de libros con bookdown

Fernández-Casal, R. y Cotos-Yáñez, T.R.

2018-10-25

Índice general

Prólogo	5
1 Introducción	7
1.1 Requisitos	7
1.2 Primeros pasos	8
1.3 Estructura del libro	8
2 Archivos de configuración	9
2.1 Encabezado de <i>index.Rmd</i> y archivo <i>__output.yml</i>	9
2.2 Archivo <i>__bookdown.yml</i>	10
3 Componentes	13
3.1 Introducción	13
3.2 Secciones y cabeceras	13
3.3 Figuras y tablas	14
3.4 Ecuaciones	14
3.5 Referencias bibliográficas	15
3.6 Entornos	16
4 Publicación	17
4.1 GitHub Pages	17
4.2 Licencia	17
A R Markdown	19
A.1 Introducción	19
A.2 Sintaxis de Markdown	20
A.3 Inclusión de código R	21
A.4 Tablas	23
A.5 Cabecera YAML	24
A.6 Extracción del código R	25
A.7 Spin	26
A.8 Extensiones RMarkdown de pandoc	26
B Pandoc	29
B.1 Introducción	29
B.2 Conversión de documentos con Pandoc	29
B.3 Pandoc y RMarkdown	30

Prólogo

Este libro es una pequeña guía sobre como emplear el paquete **bookdown** de R para la escritura de libros, incluyendo algunos detalles de configuración para la escritura en otros idiomas distintos del inglés (castellano, galego,...).

Este mismo libro ha sido escrito en R-Markdown empleando el paquete **bookdown** y está disponible en el repositorio Github: [rubenfcasal/bookdown_intro](https://github.com/rubenfcasal/bookdown_intro).

Para generar el libro (compilar) puede ser recomendable instalar la última versión de RStudio y la versión de desarrollo de **bookdown** disponible en Github. En la Sección 1.1 se detallan los requisitos necesarios.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional.

Capítulo 1

Introducción

El paquete `bookdown` (Xie, 2018a) de R (R Core Team, 2018) permite escribir libros empleando R Markdown de forma fácil. Sin preocuparse mucho por los detalles de composición se pueden crear libros en distintos formatos (HTML / LaTeX / PDF / EPUB / ...). Además de permitir emplear las extensiones Markdown de Pandoc (notas al pie de página, tablas, citas, ecuaciones LaTeX, ...), se pueden emplear extensiones Markdown para libros (leyendas de figuras y tablas, numeración y referencias cruzadas de figuras / tablas / secciones / ecuaciones / teoremas / ejemplos / ..., widgets HTML, ...).

Este libro es una pequeña guía para la escritura de libros con este paquete (incluyendo algunos detalles para la escritura en otros idiomas distintos del inglés), pero se supone que el lector ya tiene conocimientos de RMarkdown. En el Apéndice A se incluye algo de información sobre este lenguaje, pero la recomendación sería buscar fuentes adicionales de información si no se conoce, empezando por la web <http://rmarkdown.rstudio.com/> y las fuentes descritas en dicho apéndice. También recomendamos consultar fuentes externas para obtener documentación detallada sobre los distintos temas, como por ejemplo los libros (disponibles de forma gratuita en línea):

- Xie (2018a) : Authoring Books and Technical Documents with R Markdown, <https://bookdown.org/yihui/bookdown/>.
- Xie et al. (2018) : R Markdown: The Definitive Guide, <https://bookdown.org/yihui/rmarkdown/>.

Ejemplos de este tipo de libros¹ se tienen en <https://bookdown.org> (el listado completo está disponible aquí). Además de este, otros libros en castellano que se puede utilizar como ejemplo son:

- Prácticas de Simulación, disponible en el repositorio de GitHub [rubenfcasal/simbook](#) (Zip).
- R para profesionales de los datos: una introducción.

Si ya se dispone de contenido en otros formatos sobre el tema del libro, el primer paso sería su conversión a formato RMarkdown. Para ello se puede emplear Pandoc (ver Apéndice B), siguiendo los pasos descritos en la Sección B.2.

1.1 Requisitos

1. Disponer de una versión reciente de RStudio. Descargar la última versión si la versión actual es anterior a la 1.0.0.
2. Instalar el paquete `bookdown` de R:

¹Los libros de Hadley Advanced R y R packages serían ejemplos con una versión “preliminar” de este paquete.

```
# stable version on CRAN
install.packages('bookdown')
# or development version on GitHub
# devtools::install_github('rstudio/bookdown')
```

3. Para compilar los libros en pdf **bookdown** emplea XeLaTeX. En windows es recomendable instalar MiKTeX, o actualizarlo (ejecutando MiKTeX Update) si ya está instalado. En Mac OS X se puede instalar MacTeX y en Linux TeXLive².

1.2 Primeros pasos

Si se emplea RStudio (recomendado), lo más cómodo sería crear un proyecto nuevo empleando el menú *File > New Project > New Directory > Book project using bookdown*. Alternativamente se podría usar como plantilla algún libro publicado (este se puede descargar aquí). Por ejemplo, el proyecto que crea por defecto RStudio está disponible en el repositorio de GitHub `rstudio/bookdown-demo` y se puede descargar como fichero Zip. Posteriormente se pueden ir modificando (añadiendo o eliminando) los ficheros *.Rmd* correspondientes a los distintos capítulos.

Si se abre el proyecto en RStudio, se puede compilar empleando el boton *Build Book*³ en la pestaña *Build*. Por defecto se generará el libro (en los distintos formatos) en la carpeta *__book* y al terminar se abrirá automáticamente la versión HTML. El libro se generará (de forma simultánea en los formatos deseados) a partir de una serie de documentos R Markdown, cada uno correspondiente a un capítulo.

Nota: Si se produce un error, puede ser necesario⁴ eliminar manualmente los ficheros de salida (e.g. *book_filename.Rmd* o *book_filename.md*) para poder volver a compilar.

1.3 Estructura del libro

La estructura del libro se corresponde con un conjunto de directorios y ficheros, los principales serían:

- Un fichero inicial de RMarkdown *index.Rmd* con diferentes configuraciones generales (título, autor, fecha, ...) y donde se puede incluir un prólogo/preámbulo/prefacio.
- Un fichero de RMarkdown *.Rmd* para cada capítulo, definido por el título de primer nivel **#**. Los capítulos se ordenarán siguiendo el orden alfabético de los nombres de los ficheros⁵.
- Archivo *__bookdown.yml* de configuración donde se especifican parámetros opcionales para la compilación del libro.
- Archivo *__output.yml* de configuración con las opciones del formato de salida.
- Archivos *style.css* y *preamble.tex* con especificaciones de las opciones de estilo de los libros en formato HTML e LaTeX, respectivamente.
- Archivo *book.bib* con las referencias bibliográficas en formato BibTeX.

Se pueden incluir directorios adicionales, por ejemplo una carpeta *images* que contenga las imágenes estáticas y una carpeta de *data* con los archivos de datos a usar.

²Aunque el autor del paquete **bookdown** recomienda instalar TinyTeX.

³Alternativamente se puede ejecutar la función `render_book()` en el directorio del proyecto

⁴Esto debería hacerse automáticamente si se incluye la opción `delete_merged_file: true` en el fichero de configuración *__bookdown.yml*.

⁵Evitar espacios (y acentos) en los nombres de los ficheros de los capítulos.

Capítulo 2

Archivos de configuración

Nos puede interesar modificar el encabezado de *index.Rmd* y un par de archivos YAML de configuración (ver Sección A.5 para más detalles sobre este formato).

2.1 Encabezado de *index.Rmd* y archivo *__output.yml*

La cabecera YAML del primer archivo *.Rmd* del libro se puede utilizar para configurar distintas opciones:

```
---
title: "A Minimal Book Example"
author: "Yihui Xie"
date: "2018-10-25"
site: bookdown::bookdown_site
output: bookdown::gitbook
documentclass: book
bibliography: [book.bib, packages.bib]
biblio-style: apalike
link-citations: yes
github-repo: rstudio/bookdown-demo
description: "This is a minimal example of using the bookdown package to write a book.
The output format for this example is bookdown::gitbook."
---
```

Obviamente cambiaremos el título, el nombre del autor y la descripción (también nos podrá interesar cambiar las opciones de bibliografía, ver Sección 3.5, o las del repositorio GitHub, ver Sección 4.1). Por ejemplo, se puede acceder al fichero *index.Rmd* de este libro aquí .

Aunque se pueden especificar en esta cabecera otras opciones relacionadas con Pandoc, como por ejemplo opciones del formato de salida:

```
---
output:
  bookdown::gitbook:
    split_by: "section"      # Páginas por secciones (en lugar de capítulos)
    split_bib: no           # No se muestra bibliografía al final de cada página
    lib_dir: "book_assets"
  bookdown::pdf_book:
    keep_tex: yes
---
```

Normalmente estableceremos las opciones del formato de salida en el archivo de configuración YAML `__output.yml`. Por defecto se generará el libro en formato HTML (con las opciones de `bookdown::gitbook()`), en formato PDF a través de LaTeX (`bookdown::pdf_book()`) y en formato EPUB (`bookdown::epub_book()`). En este archivo también nos interesará cambiar algunas de las opciones por defecto:

```
bookdown::gitbook:
  css: style.css
  config:
    toc:
      before: |
        <li><a href=".">A Minimal Book Example</a></li>
      after: |
        <li><a href="https://github.com/rstudio/bookdown" target="blank">Published with bookdown</a></li>
        edit: https://github.com/rstudio/bookdown-demo/edit/master/%s
        download: ["pdf", "epub"]
bookdown::pdf_book:
  includes:
    in_header: preamble.tex
  latex_engine: xelatex
  citation_package: natbib
  keep_tex: yes
bookdown::epub_book: default
```

Por lo menos en el formato HTML `bookdown::gitbook`, el título del documento en la parte superior de la tabla de contenido. El fichero `__output.yml` de este libro es:

```
bookdown::gitbook:
  css: style.css
  config:
    toc:
      before: |
        <li><a href=".">Escritura de libros con bookdown</a></li>
      after: |
        <li><a href="https://github.com/rstudio/bookdown" target="blank">Publicado con bookdown</a></li>
        edit: https://github.com/rubenfcasal/bookdown_intro/edit/master/%s      # Si se publica con GitHub
        download: "pdf"                                                         # Link de descarga en pdf
bookdown::pdf_book:
  includes:
    in_header: preamble.tex
  latex_engine: xelatex
  citation_package: natbib
  keep_tex: yes
```

Más detalles sobre las opciones de salida en el libro de bookdown. Otros ejemplos en castellano de `index.Rmd` y `__output.yml`.

2.2 Archivo `__bookdown.yml`

El fichero de configuración por defecto es:

```
book_filename: "bookdown-demo"
language:
  ui:
    chapter_name: "Chapter "
```

```
delete_merged_file: true
```

Por ejemplo, nos interesará cambiar el nombre del fichero y, si se trata de un libro en castellano (galego, ...), el encabezado de los títulos de los capítulos. Además nos puede interesar cambiar los encabezados de algunos componentes como figuras y tablas, siguiendo las instrucciones de la Sección 4.5 del libro de bookdown. Por ejemplo en este libro se emplea:

```
book_filename: 'bookdown_intro'
output_dir: docs
new_session: no
language:
  label:
    fig: 'Figura '
    tab: 'Tabla '
    eq: 'Ecuación '
    thm: 'Teorema '
    lem: 'Lema '
    def: 'Definición '
    cor: 'Corolario '
    prp: 'Proposición '
    exm: 'Ejemplo '
    exr: 'Ejercicio '
    proof: 'Demostración. '
    remark: 'Nota: '
    solution: 'Solución. '
  ui:
    chapter_name: 'Capítulo '
delete_merged_file: yes
```

y puede ser descargado aquí. Además se modificó *preamble.tex* para configurar LaTeX para castellano, añadiendo:

```
\ifxetex
  \usepackage{polyglossia}
  \setmainlanguage{spanish}
  % Tabla en lugar de cuadro
  \gappto\captionsspanish{\renewcommand{\tablename}{Tabla}
    \renewcommand{\listtablename}{Índice de tablas}}
\else
  \usepackage[spanish,es-tabla]{babel}
\fi
```

Otras opciones permitirían cambiar el orden de los ficheros *.Rmd* (`rmd_files:`), o establecer y configurar la sesión de R donde se evalúa el código. Por ejemplo:

```
new_session: yes # Cada capítulo en una nueva sesión
before_chapter_script: ["script1.R", "script2.R"] # Ejecutar código antes de los capítulos
after_chapter_script: "script3.R" # y después.
```

Más detalles de las opciones de configuración en el libro de bookdown.

Capítulo 3

Componentes

3.1 Introducción

Para la elaboración de contenido se pueden emplear las componentes de RMarkdown descritas en el Apéndice A, incluyendo la sintaxis de Markdown (Sección A.2) y la inclusión de bloques de código en R (Sección A.3). Pero como se comenta en este Apéndice, RMarkdown admite extensiones adicionales (proporcionadas por Pandoc), que pueden ser de utilidad en la escritura de un libro. Por ejemplo, se pueden añadir $\text{sub}^{\text{índices}}$ y $\text{super}^{\text{índices}}$ con `sub~índices~` y `super~índices~`, notas al pie con `^[texto]`, ecuaciones y referencias bibliográficas.

El paquete `bookdown` proporciona extensiones adicionales de la sintaxis de RMarkdown especialmente diseñadas para la escritura de libros (ver p.e. la Sección 2.2 del libro de `bookdown`), además el comportamiento de algunos resultados cambia al renderizar con este paquete.

En las siguientes secciones se mostrarán algunas de estas extensiones de RMarkdown y de `bookdown` (en el Capítulo 2 del libro de `bookdown` se detallan todas las extensiones `bookdown`, incluyendo referencias de texto, bloques personalizados, HTML widgets, páginas web y aplicaciones Shiny).

3.2 Secciones y cabeceras

Como ya se comentó, en el fichero de RMarkdown `.Rmd` de cada capítulo, este está definido por el título de primer nivel (e.g. `# Título`; ver Sección A.2 para la sintaxis de los distintos niveles de cabeceras), por lo que sólo debería haber uno. Además, al renderizar con `bookdown` los capítulos y secciones se numeran automáticamente, siguiendo el orden alfabético de los ficheros.

Si no se desea numerar algún capítulo o sección, habrá que añadir `{-}` a continuación del correspondiente título (por ejemplo, en el archivo `index.Rmd` se incluyó `# Prólogo {-}`).

Para referenciar un capítulo o sección, se puede añadir una etiqueta de la forma `{#etiqueta}` a continuación del correspondiente título, después se podrá referenciar en el texto empleando `\@ref(etiqueta)`, que al renderizar producirá un enlace con la correspondiente numeración (más detalles en la Sección 2.6 del libro de `bookdown`).

Adicionalmente hay dos tipos especiales de cabeceras de primer nivel que pueden ser empleadas en `bookdown`:

- `# (PARTE) Título de la Parte {-}`: para separar capítulos en partes.
- `# (APPENDIX) Apéndices {-}`: para crear apéndices. Los capítulos posteriores se tratarán como apéndices y se numerarán alfabéticamente (A, B, C, ...).



Figura 3.1: Logo de rmarkdown (desde archivo PNG).

3.3 Figuras y tablas

En la Sección A.3.2 se comentan algunas opciones de los bloques de código que pueden ser de utilidad para las figuras (algunas se pueden establecer en `__output.yml`, ver Sección A.5). Por ejemplo, se puede establecer la opción `fig.cap` para incluir una leyenda (sin incluir sintaxis Markdown, salvo que se utilicen referencias de texto). En ese caso la gráfica se colocará en un entorno de figura (flotante en PDF/LaTeX¹), que se etiquetará y numerará automáticamente.

Para referenciar una figura, habrá que añadir el prefijo `fig:` al nombre del bloque de código. Por ejemplo, la referencia a una figura en un bloque con nombre `foo` será `\@ref(fig:foo)`.

Para incluir imágenes estáticas, en lugar de emplear la sintaxis habitual de markdown (e.g. ``), se puede llamar a la función `knitr::include_graphics()` en un bloque de código para poder emplear las extensiones bookdown. Por ejemplo, la figura 3.1 se incluyó con el siguiente código:

```
```{r rmarkdown, echo=FALSE, out.width='30%', fig.align='center', fig.cap='Logo de rmarkdown (desde archi
knitr::include_graphics("images/rmarkdown.png")
```
```

Además puede ser recomendable guardar estas imágenes en un subdirectorio (*images* en el caso de este libro). En la Sección 2.4 del libro de bookdown se tienen más detalles sobre las Figuras.

Las tablas se comportan de forma similar. Si se emplea la función `knitr::ktable()` para incluir tablas, como se describe en la Sección A.4, estas se colocarán en un entorno de tabla (flotante en PDF/LaTeX), y se etiquetarán y numerarán automáticamente. Para referenciarlas, habrá que añadir el prefijo `tab:` al nombre del bloque de código (e.g. `\@ref(tab:foo)`). Más detalles en la Sección 2.5 del libro de bookdown.

3.4 Ecuaciones

RMarkdown permite incluir expresiones matemáticas en formato LaTeX:

- En línea escribiendo la expresión latex entre dos símbolos de dolar, por ejemplo `$$\alpha, \beta, \gamma, \delta`, resultaría en $\alpha, \beta, \gamma, \delta$.
- En formato ecuación empleando dos pares de símbolos de dolar. Por ejemplo:

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

¹Se puede ajustar las posición empleando la opción de `fig.pos` (por ejemplo, `fig.pos = '!htb'`).

```
\gamma & \delta
\end{pmatrix}$$
```

resultaría en:

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

Adicionalmente en `bookdown` se pueden incluir ecuaciones numeradas (empleando directamente sintaxis LaTeX), que pueden ser referenciadas posteriormente. Por ejemplo:

```
\begin{equation}
  f\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k}
  (\#eq:ecuacion1)
\end
```

produce la siguiente ecuación (3.1):

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (3.1)$$

que puede ser referenciada con `\@ref(eq:ecuacion1)` (más detalles aquí).

3.5 Referencias bibliográficas

RMarkdown también admite bibliografía en distintos formatos, ver p.e. https://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html. Por defecto `bookdown` emplea referencias bibliográficas en formato BibTeX (lo que se describe con detalle aquí). Para especificar uno o más archivos de bibliografía *.bib*, se usa el campo `bibliography` en el fichero *Index.Rmd*. Por defecto es de la forma:

```
---
bibliography: ["book.bib", "packages.bib"]
biblio-style: "apalike"
link-citations: true
---
```

Por ejemplo, una entrada de BibTeX sería:

```
@book{xie2016bookdown,
  title={Bookdown: Authoring Books and Technical Documents with R Markdown},
  author={Xie, Yihui},
  year={2016},
  publisher={Chapman and Hall/CRC}
}
```

Si se añade al fichero *book.bib* podemos referenciarla en el texto empleando² `@xie2016bookdown` o `[@xie2016bookdown]`, resultando Xie (2016) o (Xie, 2016).

Además, en el fichero *index.Rmd* se incluye por defecto el siguiente código:

```
# automatically create a bib database for R packages
knitr::write_bib(c(.packages(), 'bookdown', 'knitr', 'rmarkdown'), 'packages.bib')
```

que genera automáticamente el archivo *packages.bib* con las referencias bibliográficas de los paquetes especificados (puede interesar añadir los paquetes empleados). En el texto RMarkdown estas referencias se incluyen con `@R-paquete` (o `[@R-paquete]`), y la referencia a R (R Core Team, 2018) con `[@R-base]`.

²En RStudio se puede instalar el “Addin” `citr` para insertar citas a referencias bibliográficas en formato BibTeX.

Tabla 3.1: Entornos en `bookdown`.

| Entorno | Nombre mostrado | Prefijo referencia |
|--------------------------|-----------------|--------------------|
| <code>theorem</code> | Theorem | <code>thm</code> |
| <code>lemma</code> | Lemma | <code>lem</code> |
| <code>definition</code> | Definition | <code>def</code> |
| <code>corollary</code> | Corollary | <code>cor</code> |
| <code>proposition</code> | Proposition | <code>prp</code> |
| <code>example</code> | Example | <code>exm</code> |
| <code>exercise</code> | Exercise | <code>exr</code> |

Pandoc generará el listado de referencias al final del documento, por lo que habría que insertar una última sección `# Bibliografía {-}` al generar el libro en formato HTML (en PDF se hará automáticamente al emplear LaTeX). Si además de HTML se emplean otros formatos, se puede añadir el código:

```
`r if (knitr::is_html_output()) '# Bibliografía {-}'`
```

al final del documento.

3.6 Entornos

El paquete `bookdown` proporciona entornos adicionales (numerados y no numerados) que pueden ser de utilidad en la escritura de un libro. En la Tabla 3.1 se muestran los tipos de entorno soportados (aunque el nombre mostrado puede variar si se cambió esta opción en el fichero `__bookdown.yml`).

Para usar uno de estos entornos hay que emplear:

```
```{example, label="foo", name="Entornos de bookdown"}
Esto es un ejemplo.
```
```

obteniéndose el Ejemplo 3.1, que se puede referenciar con `\@ref(exm:foo)` (más detalles en la Sección 2.2.2 del libro de `bookdown`).

Ejemplo 3.1 (Entornos de `bookdown`). Esto es un ejemplo.

Capítulo 4

Publicación

La “publicación” del libro web puede ser tan simple como subirlo a GitHub (estableciendo antes las opciones que se describen a continuación), aunque también bastaría con subir el contenido de la carpeta de salida `__book` a cualquier servidor web.

4.1 GitHub Pages

Para poder emplear GitHub Pages habría que cambiar el directorio de salida en el archivo `__bookdown.yml`:

```
output_dir: "docs"
```

Después de generar el libro, crear un fichero `.nojekyll` (vacío) en el nuevo directorio `docs` (GitHub procesa los sitios web con Jekyll; e.g. enlace), para lo que bastaría con ejecutar en la consola `file.create('docs/.nojekyll')`. Finalmente crear el correspondiente repositorio en GitHub, estableciendo en la configuración de “GitHub Pages” la opción de “Source” como “master branch /docs folder” (p.e. siguiendo estos pasos).

Para ir añadiendo contenido a un sitio en GitHub, se puede emplear por ejemplo RStudio (e.g. R packages by Hadley Wickham) o instalar GitHub Desktop (lo que recomendaría en Windows...).

Más información y alternativas en el libro de bookdown.

4.2 Licencia

Es recomendable incluir una licencia, por ejemplo, este libro está bajo una licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional. Sin embargo en el libro de Prácticas de Simulación se consideró una licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional, de forma preliminar.

Se puede incluir la licencia en formato texto en un archivo `LICENSE` en la raíz del proyecto.

Apéndice A

R Markdown

A.1 Introducción

Este documento se ha generado empleando Markdown y R. RMarkdown es recomendable para difundir análisis realizados con R en formato HTML, PDF y DOCX (Word), entre otros.

Markdown se diseñó inicialmente para la creación de páginas web a partir de documentos de texto de forma sencilla y rápida. Es lo que se conoce como un lenguaje de marcado ligero, tiene unas reglas sintácticas muy simples y se busca principalmente la facilidad de lectura. Posteriormente se fueron añadiendo funcionalidades, por ejemplo para incluir opciones de publicación en muchos otros formatos. La implementación original de Markdown es de John Gruber, pero actualmente están disponibles múltiples dialectos (sobre todo para publicar en gestores de contenido). RMarkdown utiliza las extensiones de la sintaxis proporcionada por *Pandoc* (ver Apéndice B), y adicionalmente permite la inclusión de código R.

Al renderizar un fichero RMarkdown se generará un documento que incluye el código R y los resultados incrustados en el documento¹. En *RStudio* basta con hacer clic en el botón *Knit*. En R se puede emplear la función `render` del paquete `rmarkdown` (por ejemplo `render("Informe.Rmd")`). También se puede abrir directamente el informe generado:

```
library(rmarkdown)
browseURL(url = render("Informe.Rmd"))
```

A continuación se darán algunos detalles sobre RMarkdown (y las extensiones Markdown de Pandoc que admite: notas al pie de página, tablas, citas, ecuaciones LaTeX, ...). Para más información (incluyendo introducciones a Markdown y RMarkdown), se recomienda consultar alguna de las numerosas fuentes disponibles, comenzando por la web oficial <http://rmarkdown.rstudio.com/>.

También se dispone de información en la ayuda de *RStudio*:

- *Help > Markdown Quick Reference*
- *Help > Cheatsheets > R Markdown Cheat Sheet*
- *Help > Cheatsheets > R Markdown Reference Guide*

Otras fuentes permiten obtener documentación más detallada, como por ejemplo:

- Web del paquete `knitr` (Xie, 2018b): <https://yihui.name/knitr>.
- Xie et al. (2018) : R Markdown: The Definitive Guide, <https://bookdown.org/yihui/rmarkdown/>.

¹Se llama al paquete `knitr` para “tejer” el código de R y los resultados en un fichero Markdown, que posteriormente es procesado con *pandoc*

- Extensiones RMarkdown de Pandoc: https://rmarkdown.rstudio.com/authoring_pandoc_markdown.html%23raw-tex.

A.2 Sintaxis de Markdown

Como ya se comentó la sintaxis de Markdown es muy sencilla. El texto no marcado se renderiza como texto normal aunque es necesario dejar una línea en blanco para insertar un salto de línea. Las principales reglas de Markdown se suman en la siguiente tabla:

| Escribir | o alternativamente | para obtener ... |
|-------------------------------------|--|---|
| # Título 1 | Título 1
===== | Los distintos niveles de encabezados |
| ## Título 2 | Título 2
----- | |
| ### Título 3 | ### Título 3 | |
| *Cursiva* | _Cursiva_ | <i>Cursiva</i> |
| **Negrita** | __Negrita__ | Negrita |
| [Enlace] (http://url.gz) | [Enlace] [1]
Más adelante...
[1]: http://url.gz | Enlace |
| ![Imagen] (rmarkdown.png) | ![Imagen] [1]
Más adelante...
[1]:http://url/b.jpg |  |
| > Blockquote
>
> --- El Autor | | Blockquote
— El Autor |
| * Lista
* Lista
+ Sub-lista | - Lista
- Lista
- Sub-lista | • Lista
• Lista
– Sub-lista |
| 1. Uno
2. Dos
a. A | 1) Uno
2) Dos
a) A | 1. Uno
2. Dos
a. A |

| Escribir | o alternativamente | para obtener ... |
|--|--|--|
| Regla horizontal | Regla horizontal | Regla horizontal |
| --- | *** | |
| `código en línea` entre comillas invertidas | | código en línea entre comillas invertidas |
| <pre> <code>``` # bloque de código 3 comillas invertidas o sangría de 4 espacios ```</code> </pre> | <pre> <code>[...]# bloque de código [...]3 comillas invertidas [...]o sangría de 4 espaci os</code> </pre> | <pre> <code># bloque de código 3 comillas invertidas o sangría de 4 espacios</code> </pre> |

Es muy recomendable dejar siempre una línea de separación entre elementos distintos consecutivos.

A.3 Inclusión de código R

Se puede incluir código R entre los delimitadores ````${r}```` y `````. Por defecto, se mostrará el código, se evaluará y se mostrarán los resultados justo a continuación. Por ejemplo el siguiente código:

```
```${r}
head(mtcars[1:3])
summary(mtcars[1:3])
```
```

produce:

```
head(mtcars[1:3])
```

```
##           mpg cyl  disp
## Mazda RX4      21.0   6  160
## Mazda RX4 Wag  21.0   6  160
## Datsun 710      22.8   4  108
## Hornet 4 Drive  21.4   6  258
## Hornet Sportabout 18.7   8  360
## Valiant         18.1   6  225
```

```
summary(mtcars[1:3])
```

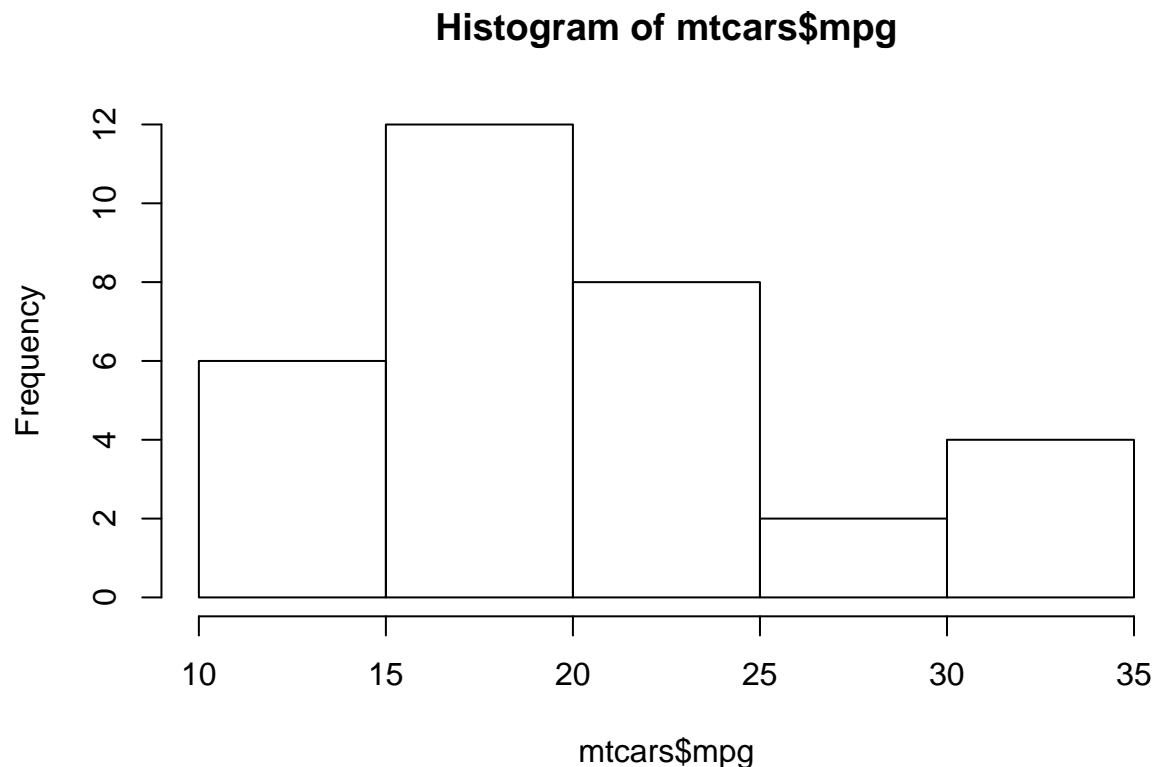
```
##           mpg           cyl           disp
##  Min.       :10.40  Min.     :4.000  Min.      : 71.1
## 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8
## Median :19.20   Median :6.000   Median :196.3
## Mean      :20.09   Mean      :6.188   Mean      :230.7
## 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0
## Max.       :33.90   Max.       :8.000   Max.       :472.0
```

En *RStudio* pulsando “Ctrl + Alt + I” o en el icono correspondiente se incluye un trozo de código.

También se puede incluir código en línea empleando ``r código``, por ejemplo ``r 2 + 2`` produce 4.

A.3.1 Gráficos

Si el código genera un gráfico, este se incluirá en el documento justo después de donde fué generado (por defecto). Por ejemplo el siguiente gráfico:



se generó empleando:

```
``{r figura1, echo=FALSE}  
hist(mtcars$mpg)  
``
```

aunque no se mostró previamente el código al haber establecido la opción ```{r, echo=FALSE}`.

A.3.2 Opciones de bloques de código

Los trozos de código pueden tener nombre y opciones, se establecen en la cabecera de la forma ```{r nombre, op1, op2}`. Para un listado de las opciones disponibles ver <http://yihui.name/knitr/options> (en la Sección 2.6 del libro de RMarkdown se incluye un resumen). En *RStudio* se puede pulsar en los iconos en la parte superior derecha del bloque de código para establecer opciones, ejecutar todo el código anterior o sólo el correspondiente trozo.

Algunas opciones sobre evaluación y resultados:

- **eval**: si `=FALSE` no se evalúa el código.
- **echo**: si `=FALSE` no se muestra el código.
- **include**: si `=FALSE` no se muestra el código ni ningún resultado.
- **message**, **warning**, **error**: oculta el correspondiente tipo de mensaje de R (los errores o warnings se mostrarán en la consola).

- **cache**: si se activa, guarda los resultados de la última evaluación y se reutilizan si no cambió el bloque de código (más detalles aquí). Puede ser de utilidad durante la redacción del documento para reducir el tiempo de renderizado (usándolo con cuidado y desactivándolo al terminar).

Algunas opciones sobre resultados gráficos:

- **fig.width**, **fig.height**, **fig.dim**: dimensiones del dispositivo gráfico de R (no confundir con el tamaño del resultado), e.g. **fig.width** = 5.
- **out.width**, **out.height**: tamaño del gráfico, e.g. **'80%'**.
- **fig.align**: **'left'**, **'center'**, **'right'**, establece la alineación.
- **fig.cap**: leyenda de la figura².
- **dev**: dispositivo gráfico de R, por defecto **'pdf'** para LaTeX y **'png'** para HTML. Otras opciones son **'svg'** o **'jpeg'**.

Para establecer valores por defecto para todos los bloques de código se suele incluir uno de configuración al principio del documento, por ejemplo:

```
```{r, setup, include=FALSE}
knitr::opts_chunk$set(comment=NA, prompt=TRUE, dev='svg', fig.dim=c(5, 7), collapse=TRUE)
```
```

A.4 Tablas

Las tablas en Markdown son de la forma:

```
First Header	Second Header
Row1 Cell1	Row1 Cell2
Row2 Cell1	Row2 Cell2
```

Por ejemplo:

| Variable | Descripción |
|----------|--|
| mpg | Millas / galón (EE.UU.) |
| cyl | Número de cilindros |
| disp | Desplazamiento (pulgadas cúbicas) |
| hp | Caballos de fuerza bruta |
| drat | Relación del eje trasero |
| wt | Peso (miles de libras) |
| qsec | Tiempo de 1/4 de milla |
| vs | Cilindros en V/Straight (0 = cilindros en V, 1 = cilindros en línea) |
| am | Tipo de transmisión (0 = automático, 1 = manual) |
| gear | Número de marchas (hacia adelante) |
| carb | Número de carburadores |

Para convertir resultados de R en tablas de una forma simple se puede emplear la función **ktable** del paquete *knitr*. Por ejemplo la Tabla A.3 se obtuvo mediante el siguiente código:

```
knitr::kable(
  head(mtcars),
  caption = "Una kable knitr"
)
```

²Si se genera un documento en PDF/LaTeX el gráfico se mostrará en un entorno flotante y se puede ajustar la posición empleando la opción **fig.pos** (por ejemplo, **fig.pos** = **'!htb'**).

Tabla A.3: Una kable knitr

| | 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 |
| Valiant | 18.1 | 6 | 225 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |

Otros paquetes proporcionan opciones adicionales: *xtable*, *stargazer*, *pander*, *tables* y *ascii*.

A.5 Cabecera YAML

En un fichero RMarkdown se puede incluir metadatos en una cabecera en formato YAML (YAML Ain't Markup Language, <https://en.wikipedia.org/wiki/YAML>), comenzando y terminando con tres guiones ---. Los metadatos de YAML son típicamente opciones de renderizado consistentes en pares de etiquetas y valores separados por dos puntos³. Por ejemplo:

```
---
title: "Creación de contenidos con RMarkdown"
author: "Fernández-Casal, R. y Cotos-Yáñez, T.R."
date: "`r Sys.Date()`"
output: html_document
---
```

Aunque no siempre es necesario, se recomienda que los valores de texto se introduzcan entre comillas (se puede incluir código R en línea, como por ejemplo ``r Sys.Date()`` para obtener la fecha actual). Para valores lógicos se puede emplear `yes/true` y `no/false` para verdadero y falso, respectivamente.

Los valores pueden ser vectores, por ejemplo las siguientes opciones son equivalentes:

```
bibliography: [book.bib, packages.bib]
```

```
bibliography:
- book.bib
- packages.bib
```

También pueden ser listas, añadiendo una sangría de dos espacios (importante):

```
output:
  html_document:
    toc: yes
    toc_float: yes
pdf_document:
  toc: yes
```

El campo `output` permite especificar el formato y las opciones de salida (por defecto se empleará la primera). Empleando este campo también se pueden especificar opciones gráficas para los bloques de código, por ejemplo:

```
output:
  html_document:
```

³La mayoría de los campos YAML son opciones que el paquete `rmarkdown` le pasa a Pandoc (ver <https://pandoc.org/MANUAL.html>).


```
fig_width: 7
fig_height: 6
fig_caption: true
```

La mayoría de los campos YAML son opciones que el paquete `rmarkdown` le pasa a Pandoc (ver documentación en el Apéndice B).

Un ejemplo adicional⁴:

```
---
title: "Creación de contenidos con RMarkdown"
subtitle: "Curso de introducción a R"
author:
- name: "Rubén Fernández Casal (ruben.fcasal@udc.es)"
  affiliation: "Universidade da Coruña"
- name: "Tomás R. Cotos Yáñez (tcotos@uvigo.es)"
  affiliation: "Universidade de Vigo"
date: "2018-10-25"
logo: rmarkdown.png
output:
  html_document:
    toc: yes # incluir tabla de contenido
    toc_float: yes # toc flotante a la izquierda
    number_sections: yes # numerar secciones y subsecciones
    code_folding: hide # por defecto el código aparecerá oculto
    mathjax: local # emplea una copia local de MathJax, hay que establecer:
    self_contained: false # las dependencias se guardan en ficheros externos
    lib_dir: libs # directorio para librerías (Bootstrap, MathJax, ...)
  pdf_document:
    toc: yes
    toc_depth: 2
    keep_tex: yes # conservar fichero latex
---
```

Como se puede deducir del ejemplo anterior, en el formato YAML podemos incluir comentarios con el carácter `#` (por ejemplo para no emplear alguna de las opciones sin borrarla del encabezado).

En el Capítulo 3 del libro de RMarkdown se tiene información detallada sobre las opciones de los distintos formatos de salida (sobre ficheros HTML aquí y sobre PDF/LaTeX aquí).

A.6 Extracción del código R

Para generar un fichero con el código R se puede emplear la función `purl` del paquete *knitr*. Por ejemplo:

```
purl("Informe.Rmd")
```

Si se quiere además el texto RMarkdown como comentarios tipo `spin`, se puede emplear:

```
purl("Informe.Rmd", documentation = 2)
```

⁴Puede ser interesante ejecutar `str(rmarkdown::html_document())` para ver un listado de todas las opciones disponibles de `html_document`

A.7 Spin

Una forma rápida de crear este tipo de informes a partir de un fichero de código R es emplear la función `spin` del paquete *knitr* (ver p.e. <http://yihui.name/knitr/demo/stitch>).

Para ello se debe comentar todo lo que no sea código R de una forma especial:

- El texto RMarkdown se comenta con `#'`. Por ejemplo:

```
#' # Este es un título de primer nivel
#' ## Este es un título de segundo nivel
```

- Las opciones de un trozo de código se comentan con `#+`. Por ejemplo:

```
#+ setup, include=FALSE
opts_chunk$set(comment=NA, prompt=TRUE, dev='svg', fig.height=6, fig.width=6)
```

Para generar el informe se puede emplear la función `spin` del paquete *knitr*. Por ejemplo: `spin("Ridge_Lasso.R")`. También se podría abrir directamente el informe generado:

```
browseURL(url = knitr::spin("Ridge_Lasso.R"))
```

Pero puede ser recomendable renderizarlo con *rmarkdown*:

```
library(rmarkdown)
browseURL(url = render(knitr::spin("Ridge_Lasso.R", knit = FALSE)))
```

En *RStudio* basta con pulsar “Ctrl + Shift + K” o seleccionar *File > Knit Document* (en las últimas versiones también *File > Compile Notebook* o hacer clic en el icono correspondiente).

A.8 Extensiones RMarkdown de pandoc

Como ya se comentó, RMarkdown utiliza la sintaxis extendida proporcionada por Pandoc. Por ejemplo, se pueden añadir sub_{índices} y super^{índices} con `sub-índices~` y `superíndices^`, y notas al pie con `^[texto]`.

Podemos incluir expresiones matemáticas en formato LaTeX:

- En línea escribiendo la expresión latex entre dos símbolos de dolar, por ejemplo `$$\alpha, \beta, \gamma, \delta`, resultaría en $\alpha, \beta, \gamma, \delta$.
- En formato ecuación empleando dos pares de símbolos de dolar. Por ejemplo:

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

resultaría en:

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

También admite bibliografía, ver p.e. https://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html. Lo más cómodo puede ser emplear un archivo de bibliografía en formato BibTeX, lo que se describe con detalle aquí. Será necesario añadir un campo `bibliography` en la cabecera YAML, por ejemplo:

```
bibliography: bibliografia.bib
csl: apa.csl # opcional
```

Suponiendo que en el directorio de trabajo están los ficheros de bibliografía *bibliografia.bib* y de estilo *apa.csl* (ver <http://citationstyles.org/>, desde donde se pueden descargar distintos archivos de estilo).

Las referencias en el texto RMarkdown se incluyen con `@referencia` o `[@referencia]`. Pandoc generará el listado de referencias al final del documento, por lo que nos puede interesar insertar una última sección `# Referencias {-}` al generar documentos HTML (en PDF se hará automáticamente al emplear LaTeX). En RStudio se puede instalar el “Addin” `citr` para insertar citas a referencias bibliográficas en formato BibTeX.

Para más detalles de las extensiones de Pandoc ver por ejemplo https://rmarkdown.rstudio.com/authoring_pandoc_markdown.html%23raw-tex o el manual de Pandoc <https://pandoc.org/MANUAL.html>.

Apéndice B

Pandoc

B.1 Introducción

Pandoc es un conversor de documentos libre y de código abierto, Pandoc puede leer archivos en distintos formatos, incluyendo:

- Distintos dialectos de Markdown
- HTML
- LaTeX
- Microsoft Word docx
- LibreOffice ODT
- EPUB

Puede convertir los documentos de entrada a muchos otros formatos, incluyendo Office Open XML, OpenDocument, HTML, Wiki markup, InDesign ICML, ebooks, OPML, y varios formatos basados en TeX (desde donde se puede producir un PDF). En la web oficial <https://pandoc.org> hay un listado completo de los formatos soportados. Pandoc también proporciona distintas extensiones de Markdown para que admita resultados más complejos.

Pandoc es una herramienta independiente de línea de comandos (sin interfaz gráfica), que se instala automáticamente con RStudio porque el paquete `rmarkdown` la emplea para generar los documentos de salida a partir de documentos Markdown (por ejemplo, en Windows en `C:\Program Files\RStudio\bin\pandoc\pandoc.exe`).

B.2 Conversión de documentos con Pandoc

La sintaxis del comando es `pandoc [opciones] [ficheros]`. Si se ejecuta `pandoc --help`, en la ventana de comandos o en la pestaña *Terminal* de RStudio, se obtiene un listado detallado de las opciones. También se puede consultar el manual de Pandoc <https://pandoc.org/MANUAL.html>.

Si Pandoc no está configurado en la ruta de búsqueda, habrá que reemplazar `pandoc` por la ruta completa al ejecutable. Por ejemplo, para emplear la versión instalada con RStudio en Windows habrá que introducir `"C:\Program Files\RStudio\bin\pandoc\pandoc"`.

Podemos emplear Pandoc para convertir contenido escrito en otros formatos a Markdown, por ejemplo:

- Un fichero word a markdown

```
"C:\Program Files\RStudio\bin\pandoc\pandoc" fichero.docx -f docx -t markdown
--extract-media . -o fichero.Rmd
```

- Un fichero LaTeX a markdown

```
"C:\Program Files\RStudio\bin\pandoc\pandoc" fichero.tex -f latex -t markdown
-o fichero.Rmd
```

- Una web a markdown

```
"C:\Program Files\RStudio\bin\pandoc\pandoc" http://url.org -f html -t markdown
-o fichero.Rmd
```

Por defecto *pandoc* produce en algunos casos un fragmento de documento (cuando el formato de salida no es markdown). Para obtener un documento independiente (e.g. un fichero HTML válido incluyendo `<head>` y `<body>`), habrá que emplear la opción `-s` o `--standalone`.

B.3 Pandoc y RMarkdown

Como ya se comentó, el paquete *rmarkdown* llama a *pandoc* para renderizar un documento RMarkdown¹, y esta llamada se muestra en la consola (o en la correspondiente pestaña de RStudio):

```
"C:/Program Files/RStudio/bin/pandoc/pandoc" +RTS -K512m -RTS Informes.utf8.md --to html4
--from markdown+autolink_bare_uris+ascii_identifiers+tex_math_single_backslash
--output Informes.html --smart --email-obfuscation none --self-contained --standalone
--section-divs --table-of-contents --toc-depth 3 --variable toc_float=1
--variable toc_selectors=h1,h2,h3 --variable toc_collapsed=1 --variable toc_smooth_scroll=1
--variable toc_print=1 --template "C:\PROGRA~1\R\R-35~1.1\library\RMARKD~1\rmd\h\DEFAUL~1.HTM"
--no-highlight --variable highlightjs=1 --variable "theme:bootstrap" --include-in-header
"C:\Users\RUBEN~1\FCA\AppData\Local\Temp\RtmpkntXD8\rmarkdown-str2084caf51da.html" --mathjax
--variable "mathjax-url:https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML"
```

Output created: Informes.html

La mayoría de los campos de la cabecera YAML de un fichero RMarkdown se traducen en las opciones de Pandoc. Por ejemplo, la cabecera:

```
---
output:
  html_document:
    number_sections: yes
---
```

(que produce la numeración de secciones y subsecciones), se corresponde con la opción `--number-sections` de *pandoc*. También se puede establecer cualquier opción de Pandoc en la cabecera YAML mediante el campo `pandoc_args`, por ejemplo:

```
---
output:
  html_document:
    pandoc_args: ["--number-offset", "4,0", "--number-sections"]
---
```

(en este caso la numeración comenzaría en 4).

¹Desde la versión 2, antes se utilizaba *knitr* y *markdown*.

Bibliografía

- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Xie, Y. (2016). *Bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC.
- Xie, Y. (2018a). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.7.
- Xie, Y. (2018b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.20.
- Xie, Y., Allaire, J., and Golemund, G. (2018). *R Markdown: The Definitive Guide*. CRC Press.