

Capítulo 27. R Markdown

Club de Lectura de R para Ciencia de Datos

Linda Cabrera Orellana* Denisse Fierro Arcos†

25 October, 2021

Abstract

Este documento es una guía de R Markdown, el cual provee un marco de escritura para ciencia de datos, que combina tu código, sus resultados y tus comentarios en prosa. Los documentos de R Markdown son completamente reproducibles y soportan docenas de formatos de salida tales como PDFs, archivos de Word, presentaciones y más. En esta caso explicaremos *pdf_document*.

Contents

1 Elementos básicos de R Markdown	2
2 Encabezado YAML	2
2.1 Formato general: encabezado YAML	2
2.2 Opciones LaTeX	3
3 Bloques de código de R o <i>chunks</i>	3
3.1 Opciones de chunks	3
3.2 Tablas	4
3.3 Código dentro de una línea	5
4 Formateo de texto	5
4.1 Expresiones Matemáticas	5
4.2 Citas	5
4.3 Insertar imágenes	6
4.4 Incluir bibliografía	6
5 Generando documentos	6
5.1 Tipos de documentos disponibles	6

*Universidad Técnica de Machala (Ecuador)

†University of Tasmania (Australia)

6 Personalizando documentos	7
6.1 Tabla de contenido	7
6.2 Cambiando variables globales	7

1 Elementos básicos de R Markdown

Contiene tres tipos importantes de contenido:

1. Un encabezado YAML (opcional) rodeado de —
2. Bloques de código de R rodeados de “`”:
3. Texto mezclado con formateos de texto simple como `#` Encabezado e *itálicas*.

2 Encabezado YAML

2.1 Formato general: encabezado YAML

YAML es la sigla en inglés de la frase “*yet another markup language*”, que significa “*otro lenguaje de marcado más*”. Este lenguaje de marcado está diseñado para representar datos jerárquicos de modo tal que sea fácil de escribir y leer para humanos. R Markdown lo utiliza para controlar muchos detalles del output.

Este encabezado determina los parámetros generales de formato para el reporte a compilar. En lenguaje computacional se entiende como los metadatos del documento; esto es, información que define el formato del archivo resultante, más no su contenido:

- **Título (title)**. Texto entre comillas que servirá de título general del documento.
- **Subtítulo (subtitle)**. Texto entre comillas que servirá de subtítulo para el título general del documento.
- **Autor (author)**. Texto entre comillas para indicar el nombre del o los autores.
- **Resumen (Abstract)**. Texto entre comillas para indicar el resumen del documento.
- **Fecha (date)**. Campo para indicar la fecha. En el texto, con la expresión `today`, se solicita que imprima la fecha actual según el calendario del sistema operativo.
- **Formato de bibliografía (csl)**. Nombre de un archivo de extensión .csl para indicar el formato de referencias en el cuerpo del texto y el listado de bibliografía al final del documento. La sigla refiere a Estilo del Lenguaje de Referencias (Citation Language Style) y permite definir si se usarán citas al estilo APA, ASA, Chicago, etc. Se adecúa a los diferentes requerimientos de referencias bibliográficas.
- Los argumentos dentro del apartado resultado (output) son los siguientes:
 - **pdf_document**: indica el formato preestablecido para compilar el documento. En este caso, se trata de un PDF. Puede ser `html_document` o `word_document`.
 - **fig_caption**: indica si las figuras deben incorporar leyendas. Es `true` por default.
 - **number_sections**: si está definido como `yes` define que se numerarán los títulos y subtítulos a lo largo del documento, de manera automática y correlativa.
 - **toc**: es la abreviación de table of contents.

2.2 Opciones LaTeX

Muchos aspectos de la plantilla LaTeX utilizada para crear documentos PDF se pueden personalizar utilizando metadatos YAML de nivel superior (tenga en cuenta que estas opciones no aparecen debajo de la sección de salida, sino que aparecen en el nivel superior junto con el título, el autor, etc.) . Por ejemplo:

- **fontsize:** Tamaño de la fuente (10pt, 11pt, or 12pt).
- **documentclass:** Clases de documentos de LaTeX (article, book, etc.)
- **linkcolor, urlcolor, citecolor:** Colores para links externos, internos y citas.

3 Bloques de código de R o *chunks*

Hay tres maneras para hacerlo:

1. Con el atajo de teclado: Cmd/Ctrl + Alt + I
2. Con el ícono “Insert” en la barra de edición
3. Tipeando manualmente los delimitadores de bloque {r} y .

3.1 Opciones de chunks

Knitr provee casi 60 opciones que puedes usar para personalizar tus bloques de código. Puedes ver la lista completa en [Opciones de bloques](#).

3.1.1 No muestra el código pero lo ejecuta y no muestra resultados (include = FALSE)

3.1.2 Muestra el código pero no lo ejecuta (eval = FALSE)

```
# Cargamos los paquetes
library(tidyverse)
library(palmerpenguins)

# Media de masa corporal de los pingüinos por especie
penguins %>%
  group_by(species) %>%
  drop_na() %>%
  summarise(
    Masa_Corporal_media = mean(body_mass_g))
```

3.1.3 No se muestra el código pero se visualizan los resultados (echo = FALSE)

```
## # A tibble: 3 x 2
##   species   Masa_Corporal_media
##   <fct>           <dbl>
## 1 Adelie        3706.
## 2 Chinstrap     3733.
## 3 Gentoo       5092.
```

3.1.4 Continúa renderizando aunque el código muestre un error. Muestra las advertencias del código (message=TRUE, warning=TRUE)

```
# Media de masa corporal de los pingüinos por especie
penguins %>%
  group_by(species) %>%
  drop_na() %>%
  summarise(
    Masa_Corporal_media = mean(sex))

## Warning in mean.default(sex): argument is not numeric or logical: returning NA
## Warning in mean.default(sex): argument is not numeric or logical: returning NA
## Warning in mean.default(sex): argument is not numeric or logical: returning NA

## # A tibble: 3 x 2
##   species   Masa_Corporal_media
##   <fct>           <dbl>
## 1 Adelie            NA
## 2 Chinstrap          NA
## 3 Gentoo             NA
```

3.1.5 Muestra el código y lo ejecuta

```
# Media de masa corporal de los pingüinos por especie
penguins %>%
  group_by(species) %>%
  drop_na() %>%
  summarise(
    Masa_Corporal_media = mean(body_mass_g))

## # A tibble: 3 x 2
##   species   Masa_Corporal_media
##   <fct>           <dbl>
## 1 Adelie        3706.
## 2 Chinstrap     3733.
## 3 Gentoo        5092.
```

3.2 Tablas

Podemos dar un mejor formato a nuestras tablas con *knitr::kable* e incluir una referencia a la tabla.

```
knitr::kable(
  penguins %>%
    group_by(species) %>%
    drop_na() %>%
    summarise(
      Masa_Corporal_media = mean(body_mass_g)),
  caption = "Promedio de masa corporal por especie"
)
```

Table 1: Promedio de masa corporal por especie

species	Masa_Corporal_media
Adelie	3706.164
Chinstrap	3733.088
Gentoo	5092.437

3.3 Código dentro de una línea

Podemos incluir código dentro del texto, por ejemplo:

Nuestra data *palmerpenguins* contiene la información de 344 pinguinos. Los pinguinos tienen una masa corporal promedio de 4201.754386 g.

Si quiero formatear los número, puedo crear una función:

```
coma <- function(x) format(x, digits = 2, nsmall = 2, big.mark = ",", small.mark = ".")
```

Ahora nuestro ejemplo quedaría:

Nuestra data *palmerpenguins* contiene la información de 344 pinguinos. Los pinguinos tienen una masa corporal promedio de 4,201.75 g.

4 Formateo de texto

Existe una hoja de referencia útil que la obtienes en Help > Markdown Quick Reference.

4.1 Expresiones Matemáticas

Ejemplo 1: Utilice la integración por partes para evaluar la siguiente integral $\int xe^x dx$.

Ejemplo 2: Utilice la integración por partes para evaluar la siguiente integral:

$$\int xe^x dx$$

Puedes encontrar más fórmulas en el siguiente [link](#).

4.2 Citas

Podemos utilizar dos formatos. El primero es sencillo, como sigue a continuación:

“Comprender las cosas que nos rodean es la mejor preparación para comprender las cosas que hay mas allá.”

— Hipatia

El segundo requiere que instales el paquete *tufte*.

“Comprender las cosas que nos rodean es la mejor preparación para comprender las cosas que hay mas allá.”

— Hipatia

4.3 Insertar imágenes



Figure 1: R Markdown de Allison Horst

4.4 Incluir bibliografía

Este Club de Lectura se basó en el libro “*R para Ciencia de Datos*” de Wickham and Grolemund [2017]. Lo puedes encontrar de forma gratuita en la web en el link [R para Ciencia de Datos](#).

5 Generando documentos

Recuerda que puedes utilizar Rmarkdown para generar varios tipos de archivo, no solo `html` como estamos utilizando en este documento. Para cambiar el tipo de archivo que produce este documento a Microsoft Word, por ejemplo, puedes cambiar `output: word_document` en el encabezado.

Tambien puedes definirlo programaticamente utilizando la funcion `render` del paquete `rmarkdown`, asi: `rmarkdown::render("nombreDelArchivo.Rmd", output_format = "word_document")`. Esto es especialmente util si tienes un script que llama a Rmarkdown para producir un documento.

Si quisieramos producir mas de un formato de archivo para un mismo documento Rmarkdown podemos modificar el YAML de la siguiente manera.

```
output:  
  html_document: default  
  word_document: default
```

Incluimos `default` al lado de cada tipo de archivo porque no vamos a modificar ninguno de sus parametros. Luego podemos crear ambos documentos utilizando la siguiente linea de codigo: `rmarkdown::render("nombreDelArchivo.Rmd", output_format = "all")`.

5.1 Tipos de documentos disponibles

- Documentos de texto
 - `html_document`

- `html_notebook` el cual no solamente crea un documento html pero tambien otros documentos de soporte, incluyendo el codigo fuente. Es una buena manera de compartir resultados de analisis.
- `word_document`, su version libre `odt_document`
- `pdf_document`
- `rtf_document` que son archivos de texto enriquecidos que pueden ser utilizados en cualquier sistema operativo
- `md_document` que es un markdown simple, o `github_document` que es un markdown especificamente disenado para ser compatible con GitHub.
- Presentaciones
 - `ioslides_presentation`
 - `slidy_presentation`
 - `beamer_presentation`
- Tableros de control (dashboards)
- Sitios web

6 Personalizando documentos

Rmarkdown es flexible y te permite personalizar tu documento tal como lo puedes hacer en un archivo de Microsoft Word. Abajo de incluimos una lista de opciones que tienes disponibles.

6.1 Tabla de contenido

Supongamos que quisieras que los nombres de cada seccion sea utilizados como una tabla de contenido. Esto lo puedes hacer modificando el encabezado YAML de la siguiente manera.

```
output:
  html_document:
    toc: true
```

Nota que `html_document` no esta en la misma linea que la opcion `output` como lo vimos anteriormente, sino que aparece en la linea abajo. Esto es porque `html_document` tiene varios parametros que pueden ser modificados, incluyendo la table de contenido (`toc`). Puedes ver otros parametros disponibles bajo `html_document` en la pestana de ayuda o usando `?html_document`.

Nota que podemos modificar la tabla de contenidos para que incluya una lista numerada de la siguiente manera.

```
output:
  html_document:
    toc: true
    toc_float: true
```

6.2 Cambiando variables globales

Si quieras aplicar un cambio al formato por defecto que aplique a todos los bloques de codigo, esto lo puedes hacer con una linea de codigo, en vez de cambiar el formato en cada bloque de manera individual. Si, por ejemplo, quieres ejecutar codigo pero no quieres que el codigo se muestre en ninguna parte del documento,

puedes usar: `knitr::opts_chunk$set(echo = FALSE)`. Recuerda que esta linea de codigo debera ir al inicio del documento y estar dentro de un bloque de codigo para que funcione.

Otra manera de esconder los bloques de codigo es haciendolo desde el encabezado YAML asi:

```
output:  
  html_document:  
    code_folding: hide
```

References

Hadley Wickham and Garrett Grolemund. *R for Data Science (2e)*. O'Reilly, 2017.