

Introducción a {RMarkdown}

Es posible que en tu trabajo tengas que presentar informes o resultados de tu análisis de datos. Tal vez te hayas encontrado guardando una y otra vez gráficos y tablas o copiando resultados de un archivo al otro hasta que el informe quedó como querías. Los archivos y el paquete **RMarkdown** vienen al rescate.

Un archivo **RMarkdown** a diferencia de un script `.R` permite incluir código en bloques o *chunks* y texto plano o con formato simple de leer usando markdown. Al poder intercalar cálculos y gráficos con su análisis o explicación, se unifica el flujo de trabajo y deja de ser necesario guardar figuras o tablas para luego insertarlas en un documento de texto. Esto es muy importante si buscamos que nuestro trabajo sea reproducible, pero además ahorra tiempo.

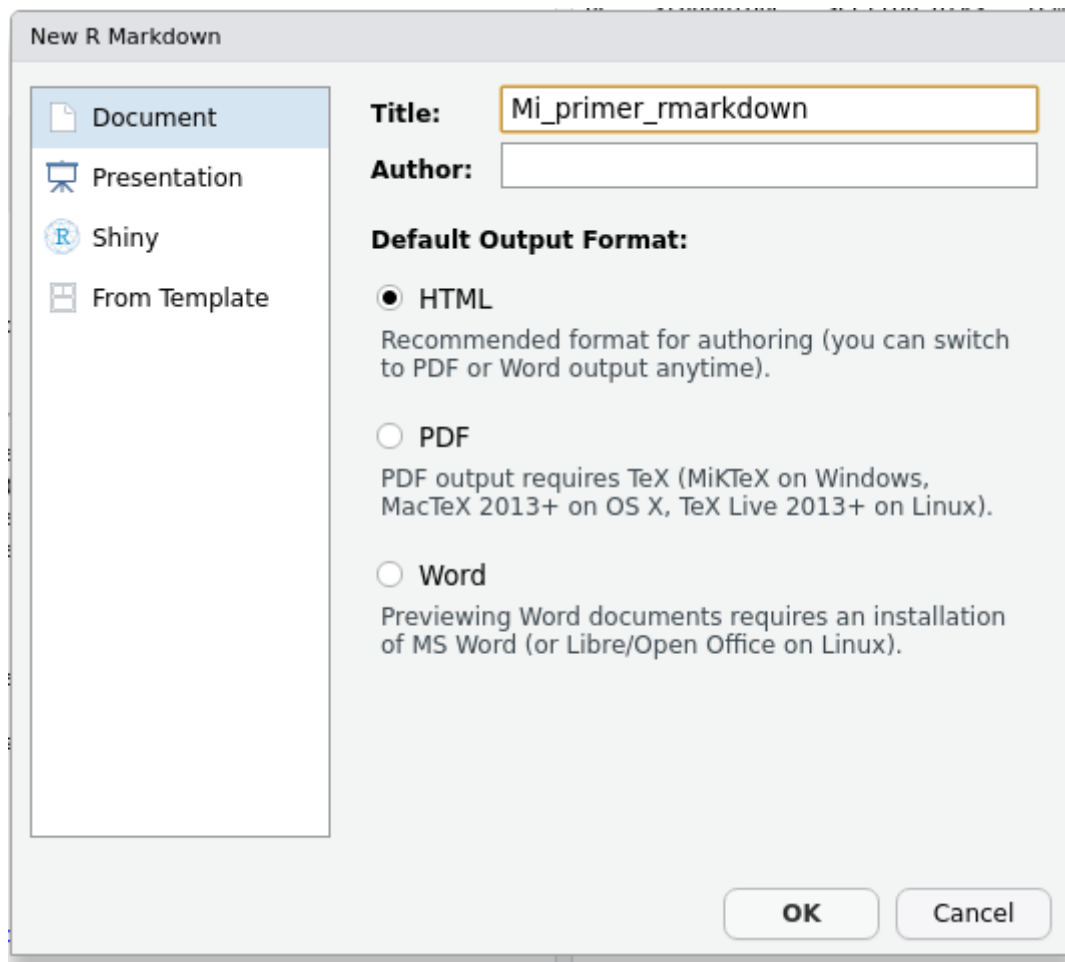
Aún si nunca necesitaras presentar un informe, trabajar con archivos de este tipo te permite volcar todo tu trabajo en un solo lugar y te asegura que cuando vuelvas a revisar mañana o el año que viene, tendrás el código y tus ideas organizadas.

Creando archivos `.Rmd`

Para crear un nuevo archivo de R Markdown usá el menú desplegable de RStudio:

File → New File → R Markdown

Y se abrirá un menú donde podés agregar el título y tu nombre. Por ahora vamos a usar el formato html como salida, charlaremos de las distintas posibilidades que ofrece más adelante.



Al aceptar, se abrirá un nuevo archivo pero que no está vacío. RStudio ofrece una plantilla a partir de la cual empezar a trabajar.

Primer desafío: Crea un nuevo archivo R Markdown

Revisá la plantilla que trae el documento. ¿Podés identificar los bloques de código?

Para generar el archivo de salida el paquete `{knitr}` (que viene de *tejer* en inglés) y algunos otros ejecutará el código en una sesión independiente de R e interpretará el texto, su formato y cualquier otra cosa que agreguemos (por ejemplo imágenes o links externos). Esto significa que nuestro archivo debe tener **todo** lo necesario para generar el análisis, si nos olvidamos de algo va a dar error.

Por esta razón es recomendable *knit*ear el archivo seguido, para encontrarnos con los errores a tiempo y de paso asegurarnos que el análisis es reproducible.

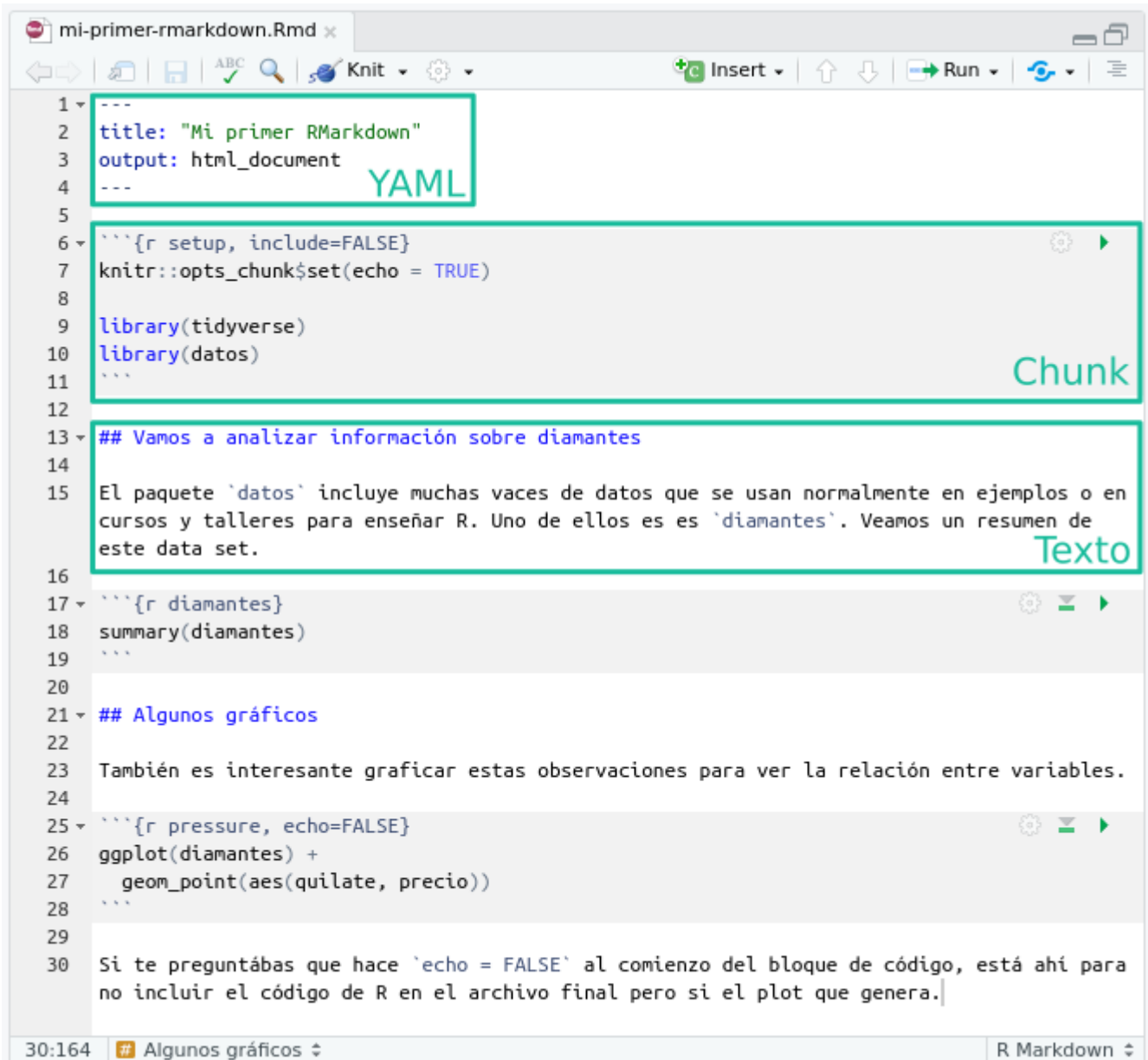
Segundo desafío: “knit

Aprovechando la plantilla de RStudio, obtené el archivo de salida en formato html > haciendo click en el botón **knit** (el que tiene un ovillo de lana y un par de > agujas!)

Estructura de un .Rmd

Cualquier archivo de este tipo tiene 3 partes principales:

- El encabezado o *yaml* que determina que pinta tendrá el archivo de salida, por ejemplo en formato html. También se puede incluir información sobre el autor, la fecha, si queremos o no una tabla de contenidos y muchas cosas más.
- El texto o textos ya que puede estar a lo largo de todo el documento. Para darle formato a los títulos o por ejemplo resaltar parte del texto usando negrita se usa **Markdown**, un *lenguaje* que a diferencia de html es legible aún si no está compilado o en su versión final.
- El código en bloques o **chunks**. Dentro de un chunk el código de R puede ejecutarse al igual que en un script normal y cualquier comentario o explicación debe tener al principio un **#** para que R lo interprete correctamente.



Podés encontrar este archivo de ejemplo en <https://github.com/paocorrales/Visualizacion/tree/main/Apuntes/mi-primer-rmarkdown.Rmd>.

Al agregar nuevas opciones en el YAML, se crean jerarquías que se identifican >> escalonando (o *indentando*) los renglones. Por ejemplo:

```
---
title: "Mi primer RMarkdown"
output:
  html_document:
    code_download: true
    toc: true
    toc_float: true
---
```

Es muy importante que no modifiques esto ya que de otra forma al hacer `knit` dará error.

Markdown

Markdown permite escribir una página web al igual que html pero de manera simple de leer. Si nuestro archivo de salida es un html, luego el paquete `knitr` se encargará de reemplazar las marcas de markdown por el código de html correspondiente. Pero como digimos más arriba también será posible generar otro tipo de archivos de salida.

Por ejemplo podrás resaltar con **negrita** usando dos asteriscos así: **`**negrita**`** o *italizada* con un asterisco de cada lado: *`*italics*`*.

También podés hacer una lista de elementos utilizando asteriscos:

```
* la negrita se consigue con dos asteriscos
* la italizada con un asterisco
* y para resaltar código se usa el acento grave `
```

o guiones medios:

```
- la negrita se consigue con dos asteriscos
- la italizada con un asterisco
- y para resaltar código se usa el acento grave `
```

Ambas listas van a tener esta pita:

- la negrita se consigue con dos asteriscos
- la italizada con un asterisco
- y para resaltar código se usa el acento grave `

Si en realidad querés una lista numerada, simplemente comenzá el renglón un número y un punto. Podrías usar siempre el mismo número, markdown se encarga del resto:

```
1. la negrita se consigue con dos asteriscos
1. la italizada con un asterisco
1. y para resaltar código se usa el acento grave `
```

Ahora la lista numerada se ve así:

1. la negrita se consigue con dos asteriscos
2. la italizada con un asterisco
3. y para resaltar código se usa el acento grave `

Podés agregar títulos con distinta jerarquía agregando `#` al comienzo. Esto además define secciones dentro del documento:

```
# Título
## El primer subtítulo
### Otro subtítulo de menor jerarquía
#### Otro más, y podría seguir!
```

Tercer desafío: Agregale texto a tu archivo

Borrá el contenido del archivo `.Rmd` que creaste (pero no el encabezado!) y probá `>` escribir algo y darle formato. Luego volvé a apretar el botón **knit** para ver el `>` resultado.

Markdown permite muchas otras cosas, por ejemplo:

Podés agregar un link a una página externa: `[texto a mostrar](http://the-web-page.com)`.

Podés incluir una imagen: `![pie de figura](http://url/for/file)`

Y también podés agregar ecuaciones en LaTeX en la misma línea (esto: `$E = mc^2$` se ve así: $E = mc^2$) o en una línea propia. Esto:

```
$$  
y = \mu + \sum_{i=1}^p \beta_i x_i + \epsilon  
$$
```

se ve así:

$$y = \mu + \sum_{i=1}^p \beta_i x_i + \epsilon$$

Podés revisar la guía rápida de Markdown desde RStudio (en inglés):

Help → Markdown Quick Reference

Bloques de código

Los bloques de código o **chunks** se definen usando 3 acentos grave al principio y al final:

```
```${r nombre-del-chunk}  
...
```
```

Y se diferencia del resto de archivo con un fondo gris.

Para insertar un nuevo chunk podés:

- Usar el botón **Insert**
- El atajo de teclado `Ctrl+Alt+I`
- Escribir a mano!

Como viste en el ejemplo, los chunks pueden tener nombre y esto es útil para identificar donde ocurren los errores al momento de *knit*ear pero también para tener una pista de lo que hace el código que incluye.

Todo lo que incluyas entre los acentos graves será interpretado por R como código e intentará ejecutarlo al *knit*ear el archivo. Pero también podemos correr el código interactivamente como en la consola, con la diferencia de que ahora el resultado (ya sea un valor, un gráfico o cualquier otra cosa) aparecerá inmediatamente por debajo del bloque de código. Para correr una línea de código, tendrás que pararte sobre esa línea y apretar:

Ctrl+Enter

Pero también podés correr el código de todo el chunk con:

Ctrl+Shift+Enter

Cuarto desafío: Sumá un chunk a tu archivo

Usando el archivo con el que venís trabajando insertá un nuevo chunk y:

1. Cargá la librería **readr**.
 2. Creá una variable que se llame **variable-prueba** y asigne un valor.
 3. Mostrá ese valor.
 4. Volvé a *knit*ear el archivo para ver el resultado
-

Texto en línea

Finalmente es posible que te encuentres mencionando resultados en el texto, por ejemplo algo así como “la media de temperatura mínima para el mes de marzo fue de 18 grados”. Y también es posible que ese valor cambie si utilizas una base de datos distinta o si luego generas un informe pero para un mes siguiente. Las chances de que te olvides de actualizar ese “18” son super altas, por eso RMarkdown también tiene la posibilidad de incorporar texto en línea.

Imaginemos que tenés una variable **temperatura** a la cual le asignamos el valor “18”:

```
temperatura <- 18
```

Para mencionarla en el texto entonces tenemos que poner el nombre de esa variable entre dos acentos graves y avisar que se trata de código de R de esta manera ``r temperatura``. Entonces si en algún momento cambia el valor de la variable, la próxima vez que *knit*es el documento se actualizará en el texto.

Fuente: Estos apuntes forman parte del curso R para clima <https://eliocamp.github.io/r-clima/> de Elio Campitelli y Paola Corrales con licencia Creative Commons Attribution-ShareAlike 4.0.