Reportes reproducibles

Table of contents

Es posible que en tu trabajo tengas que presentar informes o resultados de tu análisis de datos. Tal vez te hayas encontrando 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** o desde hace poco Quarto vienen al rescate.

RMarkdown

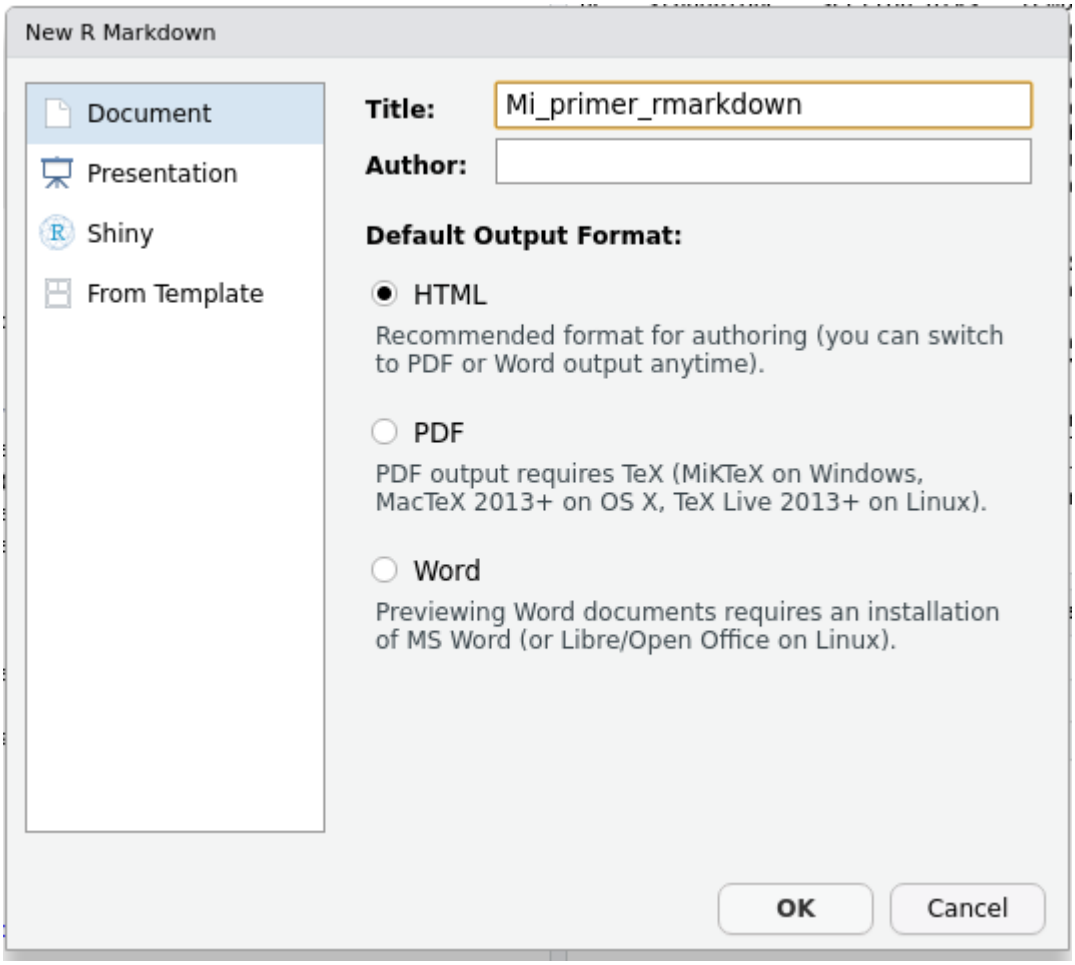
Un archivo de R Markdown (generalmente con la extensión `.Rmd`), a diferencia de un script `.R`, es un archivo de texto plano que combina código de R que genera resultados (gráficos, tablas, etc…) y el texto que lo describe. Al poder intercalar cálculos y gráficos con su análisis o explicación, se unifica 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.

Creando archivos .Rmd

En RStudio podés crear un nuevo archivo de R Markdown con el menú desplegable:

> **Instrucciones**
>
> File → New File → R Markdown

Y se abrirá un menú donde podés agregar el título de tu informe y tu nombre. Por ahora vamos a usar el formato HTML como salida, pero hay muchos otros formatos de salida posibles.



Al aceptar, se abrirá un nuevo archivo con una plantilla de ejemplo (en inglés).

Quarto

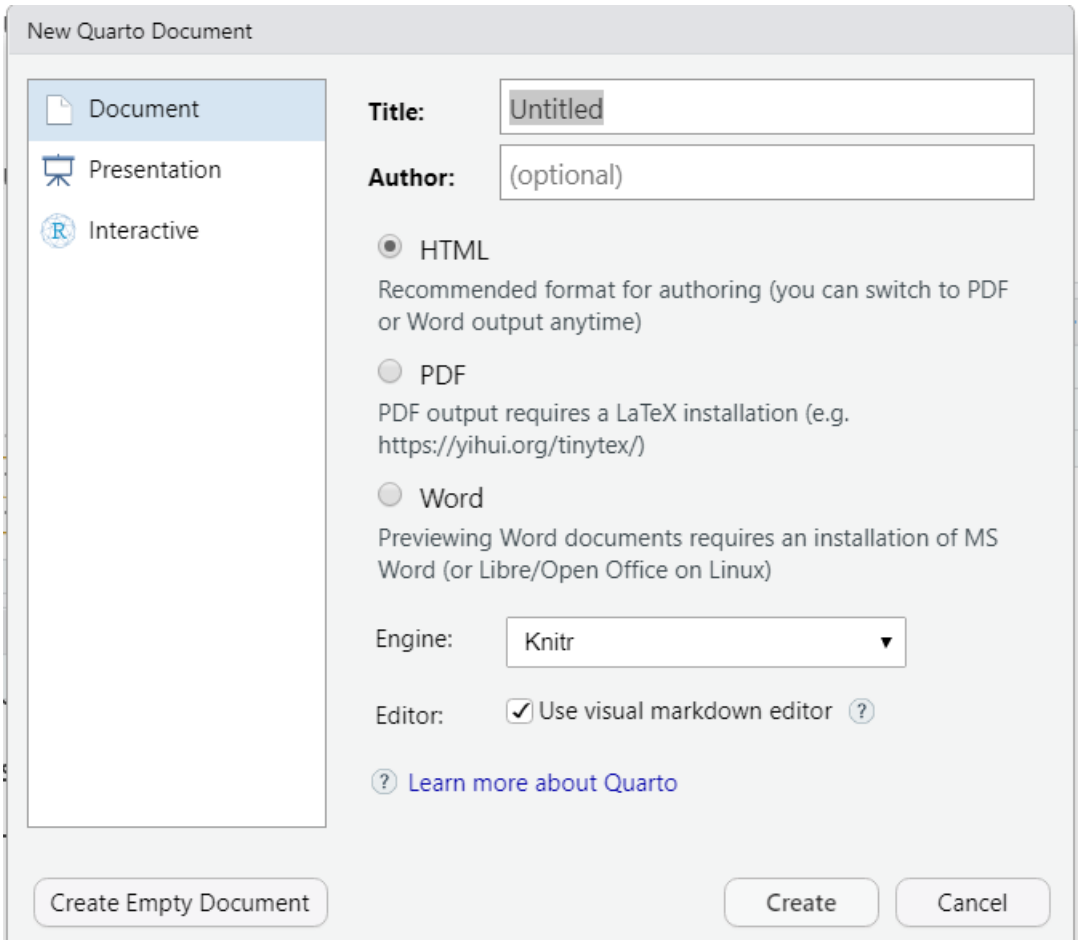
Un archivo de Quarto (generalmente con la extensión `.qmd`), a diferencia de un script `.R`, es un archivo de texto plano que combina código de R que genera resultados (gráficos, tablas, etc…) y el texto que lo describe. Al poder intercalar cálculos y gráficos con su análisis o explicación, se unifica 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. Una ventaja por sobre R Markdown es que Quarto es independiente del lenguaje de programación. En este caso estamos usando R y RStudio pero podríamos trabajar con python y notebooks de jupyter y aprovechar Quarto.

Creando archivos .qmd

En RStudio podés crear un nuevo archivo de Quarto con el menú desplegable:

> **Instrucciones**
>
> File → New File → Quarto Document

Y se abrirá un menú donde podés agregar el título de tu informe y tu nombre. Por ahora vamos a usar el formato HTML como salida, pero hay muchos otros formatos de salida posibles.



Al aceptar, se abrirá un nuevo archivo con una plantilla de ejemplo (en inglés).

Ejercicio

Primer desafío: Creá 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) 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 y si nos olvidamos de algo va a dar error.

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

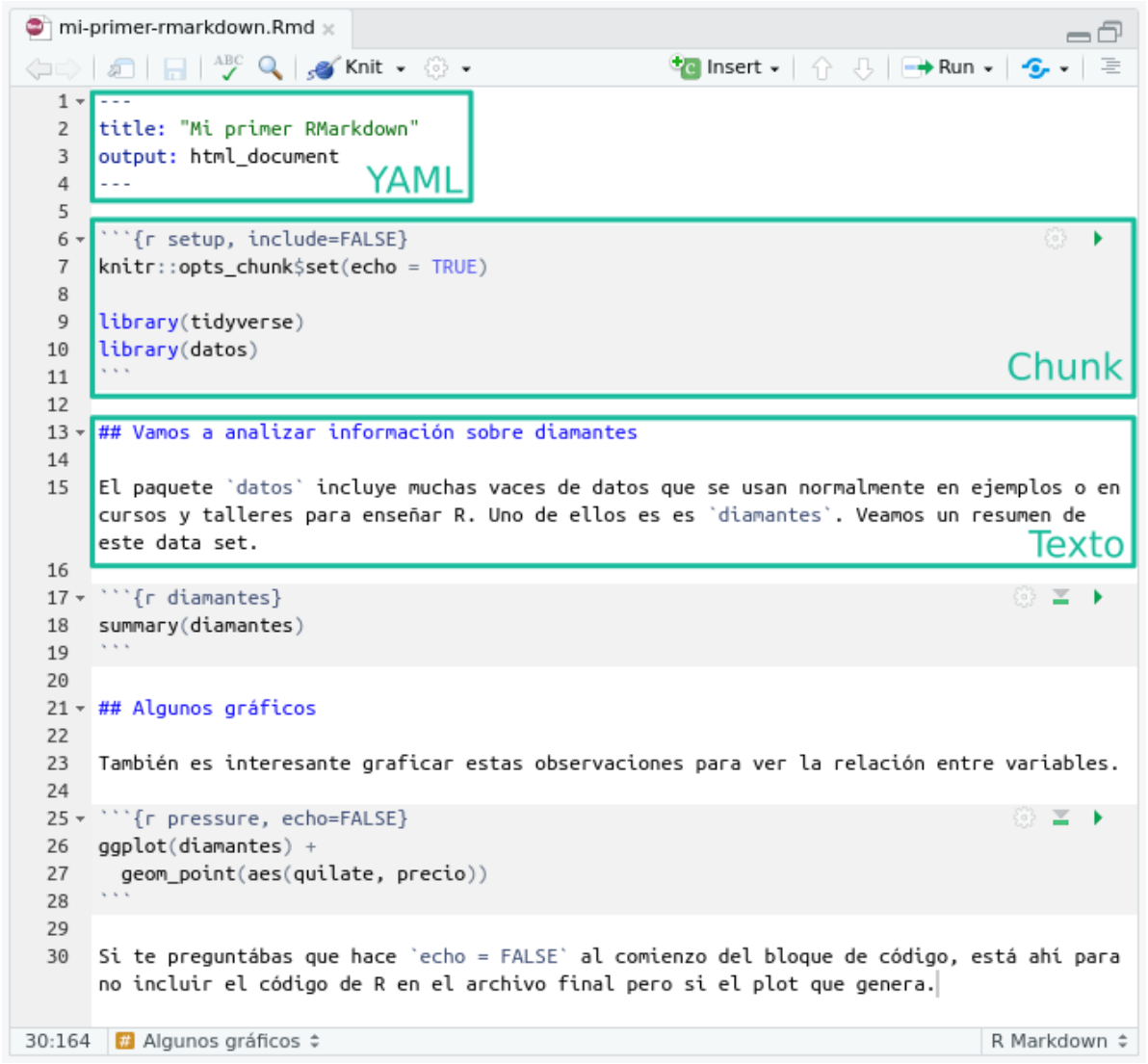
Ejercicio

Segundo desafío: renderizá tu documento de R Markdown
Aprovechando la plantilla de RStudio, obtené el archivo de salida en formato HTML haciendo click en el botón **Render** (el que tiene una flecha celeste que apunta a la derecha!).

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. Hay pequeñas diferencias entre R Markdown y Quarto.
- El **texto o prosa** 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 *chuncks***. 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.



Encabezado

El encabezado es una serie de instrucciones organizadas entre tres guiones (`---`) que determinan las propiedades globales del documento, como el título, el formato de salida, información de autoría, etc… También ahí se pueden cambiar opciones asociadas al formato de salida, como el estilo de la tabla de contenidos o índice.

Éstas propiedades se definen en un formato llamado YAML, el cual permite definir listas jerarquizadas de una forma humanamente legible. Por ejemplo:

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

define dos variables principales, “title” y “output”. “Output” a su vez contiene un elemento “htm_document”, el cual contiene tres elementos: “code_download”, “toc” y “toc_float”.

> **!** Importante
>
> Es muy importante mantener el escalonado, o *identación* de los elementos, ya que ésta define la jerarquía de cada elemento. Muchos de los errores a la hora de knitear ocurren porque el archivo tiene problemas en la identación del encabezado.

Bloques de código

El código de R que va a leer datos, analizarlos y generar figuras, tablas o números se organiza en bloques (o `chunks`) delimitados por tres acentos graves (```) y se diferencia del resto de archivo con un fondo gris. Todo lo que incluyas entre estos delimitadores será interpretado por R como código e intentará ejecutarlo al *knitear* el archivo. Cualquier resultado del código (gráficos, tablas, texto, etc…) será insertado en el documento final en el mismo orden que están en el archivo R Markdown.

> **i** Instrucciones
>
> Para insertar un nuevo chunk podés:
>
> - Usar el botón **Insert**
> - El atajo de teclado Ctrl+Alt+I
> - Escribir a mano!

El código en cada bloque se ejecuta como si fuera ejecutado en la terminal y todo resultado se muestra en el documento (ya vamos a ver formas de controlar esto). Por ejemplo, este bloque de código

````
```{r sumar}
1 + 1
```
````

va a insertar esto en el documento de salida:

```
[1] 2
```

> **Importante**
>
> Es muy importante no romper los límites de los bloques. Un problema común es accidentalmente eliminar un acento grave al final de un bloque de código y que luego el documento no knitee correctamente. Si al knitear te sale un error como "attempt to use zero-length variable name", revisá bien que todos tus bloques de código estén correctamente definidos.

Los bloques pueden tener nombre, lo cual es útil para identificar donde ocurren los errores al momento de *knitear* pero también para tener una pista de lo que hace el código que incluye.

Si bien el código se corre cuando uno knitea, cuando estés escribiendo un informe es muy cómodo ir corriendo bloques individuales interactivamente como si fuera en la consola.

Para correr una línea de código, tendrás que pararte sobre esa línea y apretar:

> **Instrucciones**
>
> Ctrl+Enter

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

> **Instrucciones**
>
> Ctrl+Shift+Enter

Los resultados van a aparecer inmediatamente debajo del bloque.

> **Ejercicio**
>
> **Tercer desafío: Sumá un chunk a tu archivo**
> Usando el archivo con el que venís trabajando insertá un nuevo chunk y:
>
> 1. Cargá el paquete readr.
> 2. Creá una variable que se llame `variable_prueba` y asignale un valor.
> 3. Mostrá ese valor.
> 4. Volvé a *knitear* el archivo para ver el resultado

Finalmente, es posible que te encuentres mencionando resultados en el texto, por ejemplo algo así como "el promedio de la variable estudiada es 3.45". 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 de que te olvides de actualizar ese "3.45" son super altas, por eso R Markdown también tiene la posibilidad de incorporar código en línea con el texto.

Si tenés una una variable `promedio` que vale "3.45":

Para mencionarla en el texto entonces escribirías:

> el promedio de la variable estudiada es ‘`r promedio`‘.

y el resultado en el documento kniteado sería

> el promedio de la variable estudiada es 3.45.

prueba: `3.45`

El texto propio del documento.

Este es el texto dirigido a las personas que van a leer el reporte. Incluirá una introducción, descripción de los datos y de los resultados; es lo que escribirías en el archivo de Word.

A diferencia de Word, el formato del texto se define usando markdown, que es un lenguaje simple que permite indicar si un texto va en negrita, cursiva, es un título, etc…usando símbolos especiales dentro del texto.

Markdown

Markdown permite escribir en texto plano pero definiendo el formato usando símbolos. Por ejemplo podés resaltar con **negrita** usando dos asteriscos así: `**negrita**` o *italizada* con un asterisco de cada lado: `*itálicas*`.

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 `
```