

Introducción al análisis de datos con R

Comunicando resultados con Rmarkdown y ejercicio final de repaso

Manuel Mejías Leiva

Universidad de Valladolid | manuel.mejias@uva.es

5 - 9 junio de 2023

Comunicando resultados con Rmarkdown

COMUNICANDO resultados: archivos .Rmd

Una de las principales **fortalezas** de **R** es la facilidad para generar informes, libros, webs, **apuntes y hasta diapositivas** (este material por ejemplo).

Para ello instalaremos antes el paquete **{rmarkdown}** que nos permitirá generar documentos **.Rmd**

```
install.packages("rmarkdown")
```

COMUNICANDO resultados: archivos .Rmd

¿Cuál son las **ventajas** de generarlos desde **rmarkdown**?

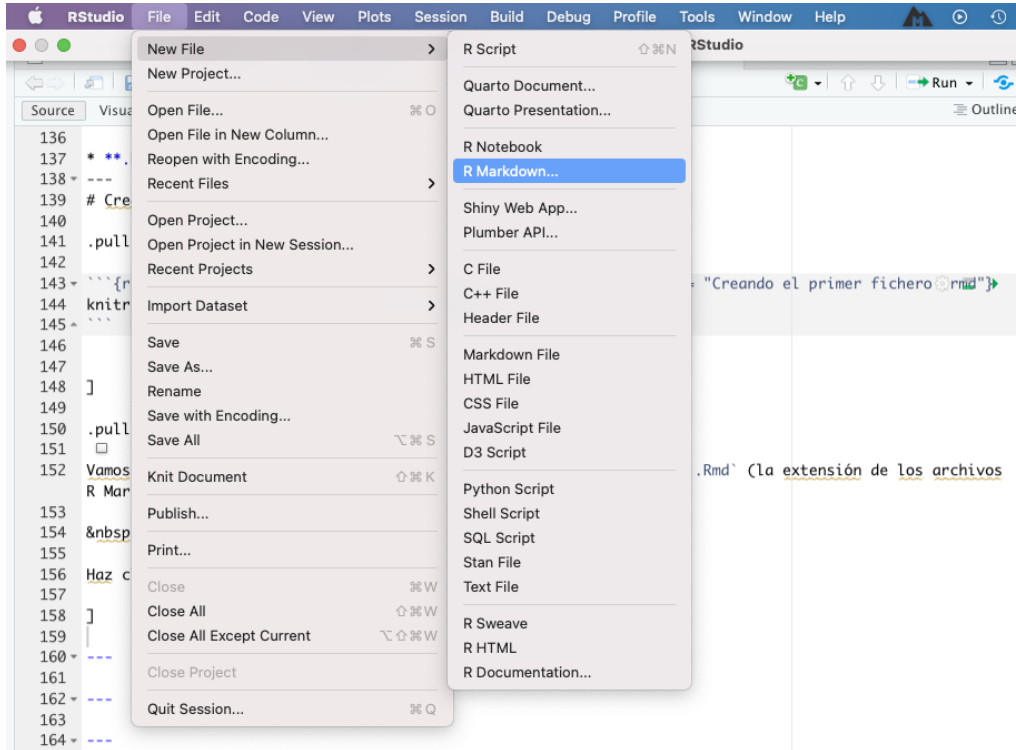
- Al hacerlo desde **RStudio**, puedes generar un informe o una presentación **sin salirte del entorno** de programación en el que estás trabajando
- Podrás analizar los datos, resumirlos y a la vez **comunicarlos**.
- Permite **integrar fácilmente código R**, de forma que no solo podremos integrar las salidas de nuestro trabajo sino también el código con el que lo hemos generado.

¿Qué es RMARKDOWN?

Una herramienta que nos permite crear de forma sencilla **documentos combinando**:

- **Markdown**: creado en 2004 por John Gruber, y de uso libre, es un «lenguaje» que nos permite crear contenido de una manera sencilla de escribir, y que en todo momento mantenga un diseño legible, con algunas de las ventajas de un HTML (si acostumbras a escribir en wordpress o blogs, seguramente hayas escrito de esta forma).
- **Matemáticas (latex)**: herramienta (lenguaje en realidad) para escribir notación matemática como x^2 o $\sqrt{2}$ (si escribes notación similar en editores de texto, seguramente sin saberlo estés usando ya latex).
- **Código** y salidas de **R**: podremos no solo mostrar el paso final sino el código que has ido realizando, con **cajitas de código** como las del manual.
- **Imágenes y tablas**.
- **Estilos** (css, js, etc).

Creando nuestro PRIMER INFORME

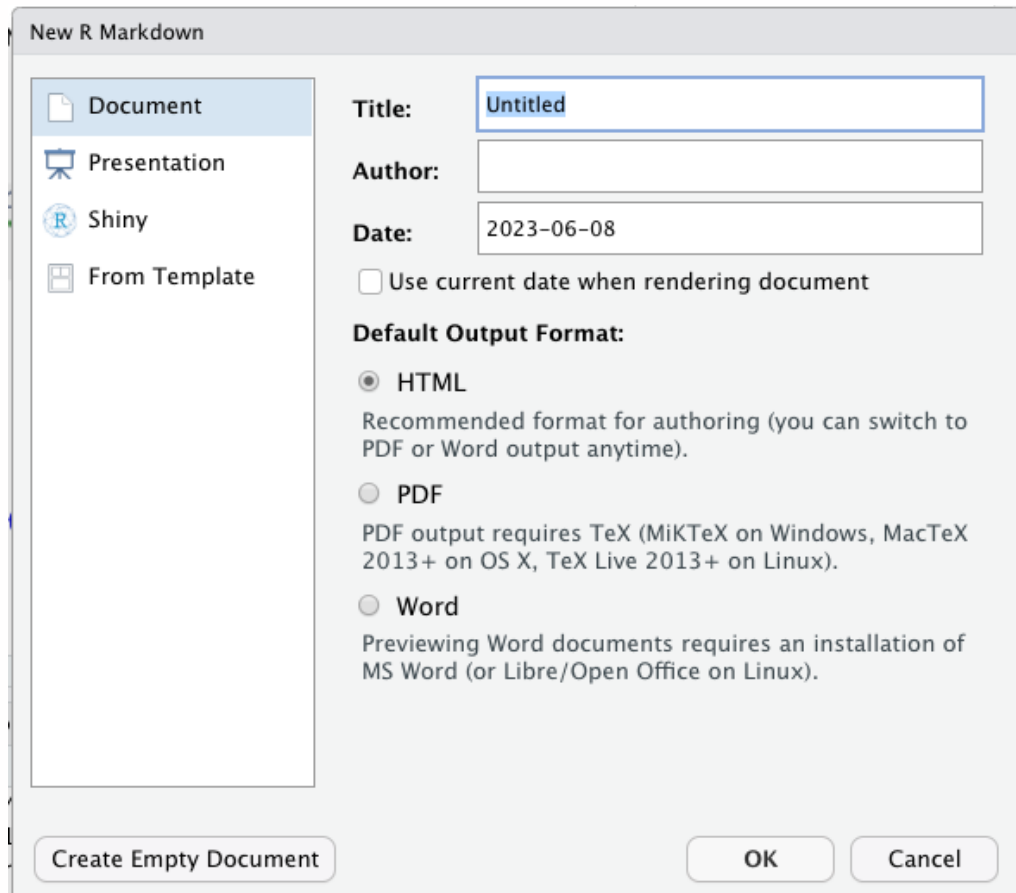


Vamos a crear el **primer fichero** con extensión **.Rmd** (la extensión de los archivos R Markdown).

Haz click en el botón **File << New File << R Markdown**.

Creando el primer fichero .rmd

Creando nuestro PRIMER INFORME



Tras hacerlo, nos aparecerán **varias opciones** de formatos de salida:

- archivo **.pdf**
- archivo **.html** (**recomendable**): documento dinámico, permite la interacción con el usuario, como una «página web»)
- archivo **.doc** (nada recomendable)

De momento dejaremos marcado el **formato HTML que viene por defecto**, y escribiremos el título de nuestro documento. Tras ello tendremos nuestro archivo **.Rmd** (ya no es un script **.R** como los que hemos abierto hasta ahora)

Creando el primer fichero .rmd

Creando nuestro PRIMER INFORME

Un fichero `.Rmd` se divide básicamente en **tres partes**

1. **Cabecera**: la parte que tienes al inicio entre `---`.
2. **Texto**: que podremos formatear y mejorar con **negritas** (escrito como `**negritas**`, con doble asterisco al inicio y final), *cursivas* (`_cursivas_`, con barra baja al inicio y final) o destacar nombres de funciones o variables de **R** (con ``R`). Recuerda que puedes añadir además ecuaciones como x^2 (he escrito `x^2`, la ecuación entre dólares).
3. **Código R**.

PRIMER INFORME: CABECERA

La cabecera están en formato **YAML**, y contiene los **metadatos del documento**: título, autor, fecha, estilos (si los tuviésemos), etc. Para probar, vamos a cambiar la cabecera que nos ha generado por defecto de la siguiente forma:

```
---  
title: "Nuestro primer Rmarkdown"  
author: "Manuel Mejías"  
date: "9/6/2023"  
output: html_document  
---
```

PRIMER INFORME: TEXTO

Solo hay una cosa **importante** a tener en cuenta en este entorno: salvo que indiquemos lo contrario, **TODO lo que vamos a escribir en el documento es texto**. No código R. Texto plano que podremos mejorar un poco con algun detalle, pero texto.

Vamos a empezar nuestro documento escribiendo por ejemplo la siguiente frase

```
Este material ha sido diseñado para el curso de Introducción al análisis de datos con R...
```

PRIMER INFORME: TEXTO



Primer informe html

Una vez que hemos escrito el texto vamos a **guardar el archivo .Rmd** haciendo click en el botón **Guardar** (yo he llamado al archivo **primer_rmarkdown.Rmd**). Tras guardar el documento, **«tejeremos» nuestro documento** haciendo click en el botón **Knit**.

Al «tejer» se nos habrá generado (seguramente en una ventana al margen) un archivo .html, que podemos incluso **abrir en nuestro navegador**. Hemos creado nuestro primer informe, obviamente vacío de momento.

PRIMER INFORME: TEXTO

```
1 ---
2 title: "Nuestro primer Rmarkdown"
3 author: "Manuel Mejias"
4 date: "9/6/2023"
5 output: html_document
6 ---
7
8 Este material ha sido diseñado para el curso de Introducción al análisis de datos con R. El curso se
9 imparte en la [Facultad de Educación de Segovia](https://educasg.uva.es/)
10
```

Tuneando nuestro primer informe html

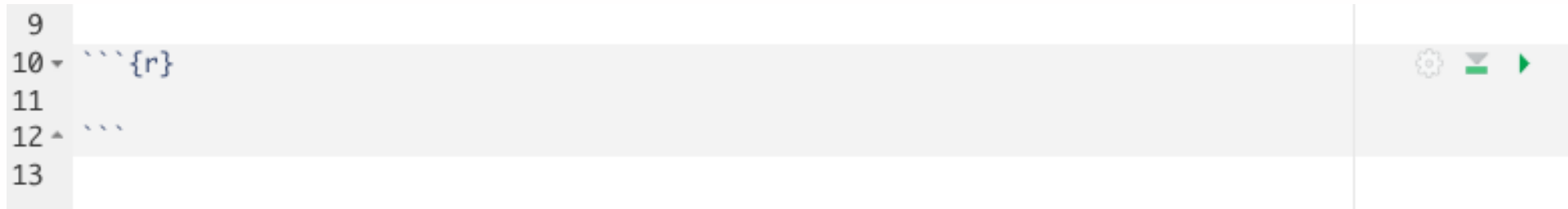
Vamos a **mejorar** un poco el texto haciendo lo siguiente:

- Vamos a añadir **negrita** al nombre (poniendo ****** al inicio y al final).
- Vamos a añadir *cursiva* a la palabra **material** (poniendo al inicio y al final).
- Vamos a añadir un enlace **https://educasg.uva.es/**, asociándolo al nombre de la Universidad. Para ello el título lo ponemos entre corchetes y justo detrás el enlace entre paréntesis **[«Facultad de Educación de Segovia»](https://educasg.uva.es/)**

PRIMER INFORME: CHUNKS de R

Para añadir **código R** debemos crear nuestras **cajas de código** llamadas **chunks**: altos en el camino en nuestro texto markdown donde podremos incluir **código**. Para incluir uno deberá de ir encabezado de la siguiente forma.

```
9
10 ▾ ```{r}
11
12 ▴ ```
13
```



Encabezado/final del chunk

PRIMER INFORME: CHUNKS de R

Dentro de dicha **cajita** (que tiene ahora **otro color** en el documento) escribiremos **código R**, como lo veníamos haciendo hasta ahora. Vamos por ejemplo a **definir dos variables** y su suma de la siguiente manera, escribiendo dicho código en nuestro **.Rmd** (dentro de ese chunk)

```
# Código R
x <- 1
y <- 2
x + y
```

```
13
14 {r}
15 # Código R
16 x <- 1
17 y <- 2
18 x + y
19
20
```

Primer chunk con código

```
## [1] 3
```

PRIMER INFORME: CHUNKS de R

Como ves dentro de esos *chunks* puedes **comentar código** con `#` (ahora veremos que hace `#` fuera de esas cajas de código). Tras hacerlo tejemos de nuevo y obtenemos ahora un documento que tiene una caja de código y su salida.

Nuestro primer Rmarkdown

Manuel Mejías

9/6/2023

Este *material* ha sido diseñado para el curso de **Introducción al análisis de datos con R**. El curso se imparte en la [Facultad de Educación de Segovia](#)

```
# Código R  
x <- 1  
y <- 2  
x + y
```

```
## [1] 3
```

PRIMER INFORME: ORGANIZANDO

Con todo incluido en el documento podemos **dividirlo en secciones y subsecciones**. Para ello usaremos la sintaxis de markdown, poniendo **almohadillas**: una `#` para secciones, `##` para subsecciones, `###` para subsubsecciones, etc. Por ejemplo, vamos a

- Hacer una sección principal que sea `# Primer informe`
- Tras ello añadiremos la parte de texto.
- Creamos una subsección que se titule `## Chunks de código` donde incluiremos los dos chunks que tenemos hasta ahora.

Además podemos incluir tras el título (y entre llaves `{}`) **etiquetas** (con `{#etiqueta}`) para luego **referenciar dichas secciones** en el documento.

También podemos organizar nuestro código **creando listas**, usando `*` como ítems.

PRIMER INFORME: PERSONALIZAR

En cada chunk aparece una **botón de play**: pulsándolo podemos tener la **ejecución y salida** de cada chunk en nuestro **.Rmd**, sin tener que esperar a «tejer» (con Knit) todo el documento para ver lo que vamos ejecutando.

Además podemos **incluir código R dentro de la línea de texto** (en lugar de mostrar el texto x ejecuta el código R mostrando la variable).

PRIMER INFORME: PERSONALIZAR

Los chunk podemos **personalizar su salida** con algunas opciones, pasándolos como argumentos dentro de las llaves ({r etiqueta, ...}).

- `include = FALSE`: **ejecuta código** pero **no se muestra (ni resultados)** en la salida.
- `echo = FALSE`: **ejecuta código** y se **muestra resultado** pero **no el código** en la salida.
- `eval = FALSE`: se **muestra el código** pero **no se ejecuta** en la salida final.
- `message = FALSE`: se **ejecuta el código** pero **no se muestran mensajes** de salida que tendríamos en consola.
- `warning = FALSE`: **ejecuta código** pero **no se muestran warning**.
- `error = TRUE`: se **ejecuta el código** pero permite ejecutar el código **con errores** mostrando los mensajes de error.

Estas opciones podemos aplicarlas chunk a chunk o fijar los parámetros de forma global con `knitr::opts_chunk$set()` (dentro de un chunk), pasándole como argumentos dichas opciones (por ejemplo, `knitr::opts_chunk$set(echo = FALSE)`).

Ejercicio final de repaso



Enunciado

El objetivo de este ejercicio es investigar si **las circunstancias socioeconómicas durante la infancia de los adultos encuestados tienen influencia en la obtención de estudios universitarios en la edad adulta**. Para tratar de dar respuesta a este objetivo general, utilizaremos la base de datos de la *Encuesta de Condiciones de Vida (ECV)* del año 2019, llevada a cabo por el Instituto Nacional de Estadística (INE).

En este ejercicio abordaremos la gran parte de los temas tratados a lo largo del curso, que incluyen la importación de datos, la limpieza de los mismos, el análisis exploratorio, la visualización y la modelización mediante modelos de regresión.

Se trabajarán con las siguientes variables de la ECV (entre paréntesis se muestran los nombres con los que aparecen en la hoja del cuestionario):

- **Nivel educativo terminado por el encuestado/a** (PE040)
- **Género** (RB090)
- **Edad** (RB080)
- **Situación económica del hogar cuando el adulto era adolescente** (PT190)

IMPORTANTE: abre el cuestionario para tener presente las características de las variables incluidas en la base de datos.



Apartado 1:

A continuación se realizarán los siguientes pasos:

- Carga las librerías necesarias para el análisis: `tidyverse`, `sjPlot` y `ggeffects`.
- Define el directorio de trabajo.
- Importa los datos.
- Realiza un primer análisis exploratorio: describe la naturaleza de las variables y muestra las filas y columnas de la base de datos con `glimpse`. Escribe en el script usando `#` el tipo de variables (numérica, factor, character, etc), el número de columnas y el número de filas que tiene la base de datos.
- Utiliza la función `summary` para obtener un resumen estadístico inicial de las variables. Por ejemplo, escribe el número de valores perdidos (NA) que tiene la variable de situación económica del hogar durante la infancia.



Apartado 2:

En este apartado se llevarán a cabo las siguientes tareas de limpieza de datos:

- Filtra los datos por edad, considerando únicamente a los encuestados de entre 25 y 59 años. Realiza este paso con la función **filter**.
- Elimina todos los valores perdidos de la base de datos con la función **drop_na**. En este ejercicio se trabajará únicamente con los casos completos en todas las variables.
- Usar la función **mutate** o **transmute** (esta última opción es más cómoda) para limpiar las variables:
 - Crea una nueva variable llamada **"uni"** que tenga el valor 1 para aquellos encuestados que han conseguido estudios universitarios y el valor 0 para el resto de encuestados que han conseguido estudios menores a los universitarios.
 - Añade etiquetas a la variable "género" (1 = hombre; 2 = mujer).
 - Recodifica la variable "edad" en los siguientes grupos: 25-29; 30-34; 35-39; 40-44; 45-52.
 - Añade las categorías a la variable "situación económica durante la infancia" (mala, muy mala, moderadamente mala, etc).

CONSEJO: sería útil usar **if_else** o **case_when** para recodificar las variables.



Apartado 3:

En este apartado se pide crear un gráfico de columnas que muestre en el eje X la variable "situación económica durante la infancia" y en el eje Y el porcentaje de encuestados con educación universitaria. Para ello, se calculará previamente el porcentaje de encuestados con educación universitaria según la situación económica del hogar durante su infancia y se asignará a un nuevo objeto llamado **"dfplot"**.

Para crear el gráfico de columnas que muestra el porcentaje de encuestados con educación universitaria según la situación económica del hogar durante su infancia, se puede seguir el siguiente trozo de código:

```
dfplot <- datos %>% #Añade a un nuevo objeto la nueva base de datos que vas a crear
  drop_na(variable1, variable2) %>% # Elimina valores perdidos
  group_by(variable1, variable2) %>% # Agrupa por las variables
  summarise(n = n()) %>% # Calcula el número de casos en cada combinación de variables
  mutate(porcentaje = (n / sum(n)) * 100) # Calcula el porcentaje y crea una nueva variable

# Completa el siguiente trozo de código para crear el gráfico de columnas
___ %>%
  filter() %>% #quédate solo con aquellos/as que tienen estudios universitarios
  ggplot(aes(x = ___, y = ___)) +
  geom_?() +
  labs()
```



Apartado 4:

En este apartado, se procederá a crear un modelo de regresión logística para analizar la asociación de las variables socioeconómicas con el logro de estudios universitarios por parte de los encuestados. Se considerará la **variable dependiente "uni"**, que toma el valor 1 si el encuestado ha conseguido estudios universitarios y 0 en caso contrario. Las **variables independientes** clave serán la **situación económica durante la infancia, la edad y el género**.

A continuación, se detallan los pasos a seguir:

- Crea un modelo de regresión logística. $VD = uni \sim VI = \text{resto de variables}$. La función necesaria para modelizar una regresión logística en R es `glm()`.
- Modifica las categorías de referencia de las variables si consideras conveniente. Esto puedes hacerlo con `relevel`.
- Muestra los resultados del modelo e interprétalos (puedes servirte de la función `report`). CUIDADO! Antes tienes que cargar la librería `report`.
- Calcula las probabilidades predichas para la variable situación económica del hogar (puedes usar `plot_model` o `ggeffects`). Por último, interpreta el gráfico que se genera.