

Investigación reproducible con R

Informes dinámicos con R Markdown

M. A. Ibáñez y J.C. Martínez Ávila
Dpto Economía Agraria, Estadística y Gestión de Empresa
ETSIAB 

Objetivos de la sesión

- Entender qué es investigación reproducible
- Conocer R Markdown como herramienta clave
- Crear informes dinámicos y automatizados
- Aplicar buenas prácticas reproducibles

¿Por qué investigación reproducible?

Problemas habituales:

- Resultados difíciles de reproducir
- Copiar/pegar entre salidas y Word. Propenso a errores
- Versiones inconsistentes de informes

¿Podría regenerar este informe dentro de 2 años?

¿Qué es investigación reproducible?

Un análisis reproducible es aquel que otra persona (o tú en 6 meses) puede ejecutar y obtener los mismos resultados.

Datos + código + documentación = mismos resultados

Principios:

- Transparencia
- Trazabilidad
- Automatización
- Reutilización

El ecosistema de R para reproducibilidad

- R → análisis utilizando código
- RStudio → entorno de programación con R (también Python, JavaScript, C++)
- R Markdown → genera documentos dinámicos

¿Qué es R Markdown?

R + Markdown + Pandoc

- Markdown es un lenguaje de marcado ligero
- R Markdown combina:
 - Texto
 - Código
 - Resultados
- Pandoc convierte Markdown a otros formatos: HTML, pdf, Doc
- Un solo documento reproducible (fichero.Rmd)

fichero.Rmd → fichero.md → fichero.html
fichero.pdf
fichero.docx

Formatos de salida

- HTML
- PDF
- Word
- Presentaciones
- Paneles de control

Anatomía de un R Markdown

Un documento R Markdown es un documento en formato texto plano con extensión .Rmd

El documento contiene:

- ① Encabezado YAML
- ② Texto (con sintaxis Markdown)
- ③ Bloques de código R

Todo integrado en un único documento

Encabezado YAML

- Son metadatos que controlan el formato de salida
- Se coloca al principio del documento
- Van entre dos líneas con tres signos menos (---)

```
---
```

```
title: "Informe de análisis"
author: "Investigador/a"
date: "2026-01-12"
output: html_document
```

```
---
```

Section 1

Texto con Markdown

Formato en línea

- *texto en itálica* utilizando *text en itálica*
- **texto en negrilla** utilizando **text en negrilla**
- NH_4^+ utilizando $\text{NH}^{\sim 4 \sim ^+}$
- Enlaces, RStudio utilizando
[Rstudio] (<https://www.rstudio.com>)
- Imagen utilizando ! [alt text or image title] (path/to/image)



Secciones y subsecciones

Utilizando el símbolo # (numeral o almohadilla)

```
# Primer nivel
```

```
## Segundo nivel
```

```
### Tercer nivel
```

Listas

- Primer elemento
- Segundo elemento
- Tercer elemento
 - Primer elemento
 - Segundo elemento
- Cuarto elemento

La salida:

- Primer elemento
- Segundo elemento
- Tercer elemento
 - Primer elemento
 - Segundo elemento
- Cuarto elemento

Listas numeradas

1. Primer elemento
2. Segundo elemento
3. Tercer elemento
 - Primer elemento
 - Segundo elemento
4. Cuarto elemento

La salida:

- ① Primer elemento
- ② Segundo elemento
- ③ Tercer elemento
 - Primer elemento
 - Segundo elemento
- ④ Cuarto elemento

Código R

En un documento R Markdown hay dos formas de introducir código R

- En un bloque de instrucciones (code chunk)
- Insertado en una línea de texto

Code chunk

```
```{r, chunck-label,chunk-options}
```

Líneas de código R

```
```
```

Code chunck

Para insertar un bloque de instrucciones en Rstudio

- Code/Insert Chunk (Alt+Ctrl+I)
- Hay una gran cantidad de opciones para los chunck que pueden verse en <https://yihui.name/knitr/options>.

Por ejemplo, el code chunck

```
```{r, label=suma, eval=TRUE, echo=TRUE}
a <- 4
b <- 5
...```

```

El resultado:

```
a <- 4
b <- 5
```

## código R en una línea

- La suma de  $a + b$  es `r a+b`

El resultado de la línea al compilar el documento es: La suma de  $a + b$  es 9

# Figuras

- Las figuras producidas por el código R se colocarán inmediatamente después del fragmento de código a partir del cual se generan.

Por ejemplo, el chunk:

```
```{r, fig.cap="Histograma de x"}  
x <- rnorm(n=1000)  
hist(x,main="")  
```
```

Genera la siguiente imagen.

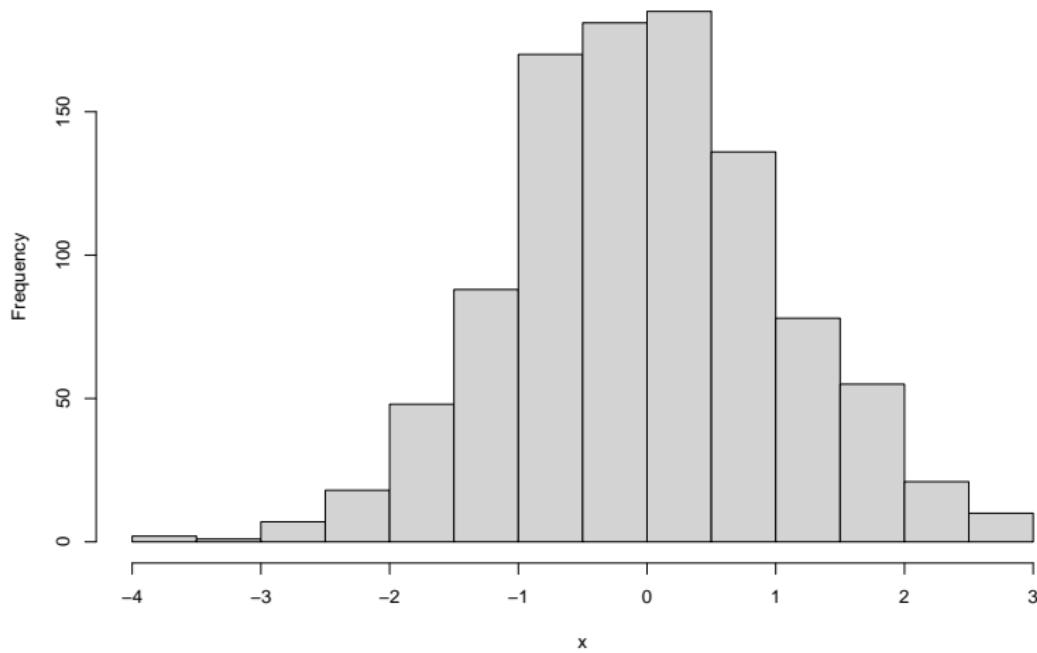


Figure 1: Histograma de x

# Tablas

- La forma más sencilla de incluir tablas es mediante `knitr::kable()`
- Solo para objetos que son marco de datos y matrices
- Crea tablas para salidas HTML, PDF y Word

- Los títulos de las tablas se pueden incluir pasando el título a la función

Por ejemplo, el chunk

```
```{r}
knitr::kable(cars[1:5],caption="5 primeras observaciones
del marco de datos `cars`")
```

```

Genera la siguiente tabla

Table 1: 5 primeras observaciones del marco de datos cars

| speed | dist |
|-------|------|
| 4     | 2    |
| 4     | 10   |
| 7     | 4    |
| 7     | 22   |
| 8     | 16   |

## Enlaces externos

- Initial steps toward reproducible research.  
<https://kbroman.org/steps2rr/>

Karl Broman es profesor del Departamento de Biostatistics & Medical Informatics de la Universidad de Wisconsin-Madison

- R Markdown: The Definitive Guide  
<https://bookdown.org/yihui/rmarkdown/>. Yihui Xie, J. J. Allaire, G. Grolemund. 2023
- R Markdown  
Cookbook <https://bookdown.org/yihui/rmarkdown-cookbook/>. Yihui Xie, C. Dervieux, E. Riederer. 2025

Yihui Xie <https://yihui.org> es ingeniero de software en RStudio  
<https://www.rstudio.com>.

Es autor de varios paquetes de R entre ellos **knitr**. También es coautor del paquete **rmarkdown**.