

RProjects y RMarkdown

Alejandra G.Cabanillas

Que son los *projects* de Rstudio?

- ▶ Una manera de organizar tu *workflow*.
- ▶ Están automáticamente asociados con el *working directory*
- ▶ Así que podemos dejar **setwd()** descansar en paz y evitar problemas de archivos que no están en su sitio.

El *tweet* y el *post* que me convirtieron a este método.

Estructura

Cada proyecto tiene su propio:

- ▶ *directory*
- ▶ *workspace*
- ▶ *history*
- ▶ *source documents*

Video de como crear RStudio *Projects*.

Ventajas

- ▶ Los análisis de diferentes proyectos se mantienen completamente separados.
- ▶ Puedes tener abiertas varias sesiones de R a la vez sin que interfieran entre ellos.
- ▶ El uso de *relative paths* (./) automáticos.
- ▶ Las direcciones de los archivos se vuelven relativas en vez de absolutas facilitando trabajar en un proyecto desde varios ordenadores.

Como crear un *Project* nuevo

New Project

Create Project

-  **New Directory**
Start a project in a brand new working directory >
-  **Existing Directory**
Associate a project with an existing working directory >
-  **Version Control**
Checkout a project from a version control repository >

Cancel

 Project: (None) ▾

 **New Project...**

 **Open Project...**

Open Project in New Session...

Close Project

first_project	
QUADstatR.io	
Rbook	
first_repo	

Imagen de @Allison Horst



¿Que es RMarkdown?

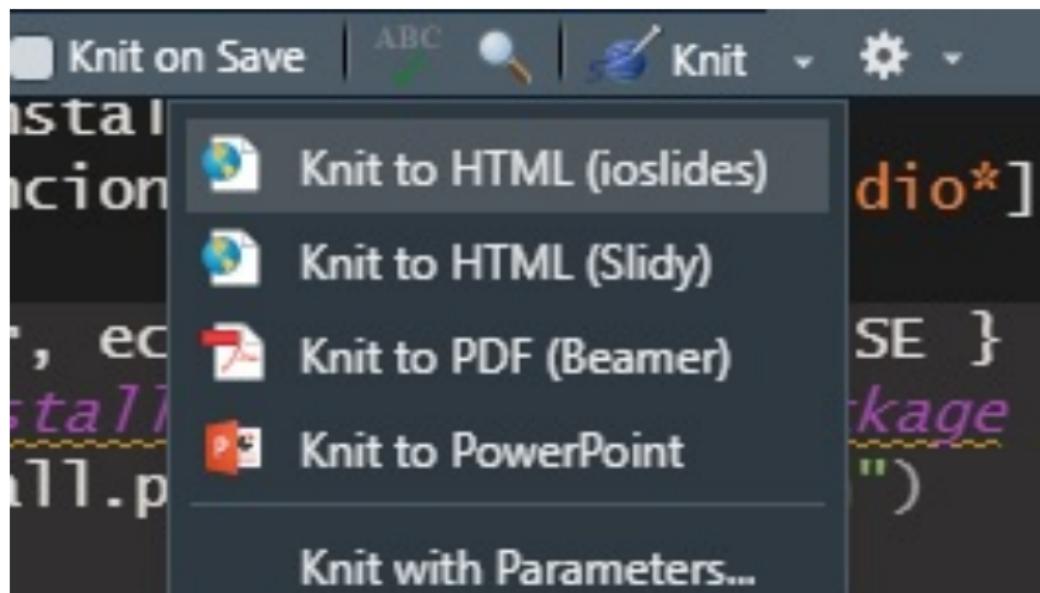
- ▶ Es un formato de archivo para crear documentos dinámicos.
- ▶ Es un formato de sintaxis simple que permite crear archivos de PDF, MS Word, MS Powerpoint y HTML.
- ▶ Permite incluir código de R (*chunks*).
- ▶ Los archivos RMarkdown son la fuente para documentos reproducibles con código y figuras integrados.

Tutoriales de RMarkdown Reportes en Markdown(Español),
Workshop en detalle de como utilizarlo, partes basicas,

Funciona en dos partes:

► knit

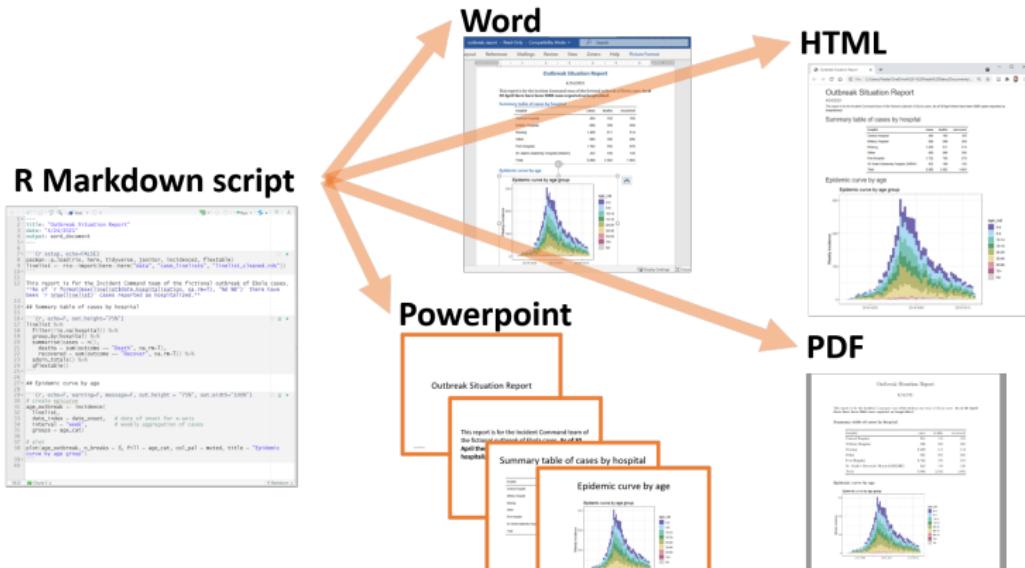
- ▶ Ejecuta el código dentro del documento.
- ▶ Utiliza el paquete **knitr**
- ▶ El *output* del código se vera justo debajo del código que lo ha producido.
- ▶ Significa que cuando cambias los datos de origen de la gráficas, estas se actualizan automáticamente.



Funciona en dos partes:

► convert

- ▶ Puedes convertir tu archivo a muchos formatos diferentes fácilmente.
 - ▶ *R*markdown utiliza el programa **pandoc** (esta automáticamente dentro de la instalación de rmarkdown).
 - ▶ Significa que puedes compartir tu documento en una variedad de formatos sin tener que cambiar el original.



Instalacion

- ▶ Funciona dentro de *RStudio*

```
# install the rmarkdown package from CRAN in R  
install.packages("rmarkdown")
```

- ▶ Para documentos de PDF necesitas *LaTeX*. La version de R es *TinyTeX*

```
tinytex::install_tinytex()  
# to uninstall TinyTeX, run  
# tinytex::uninstall_tinytex()
```

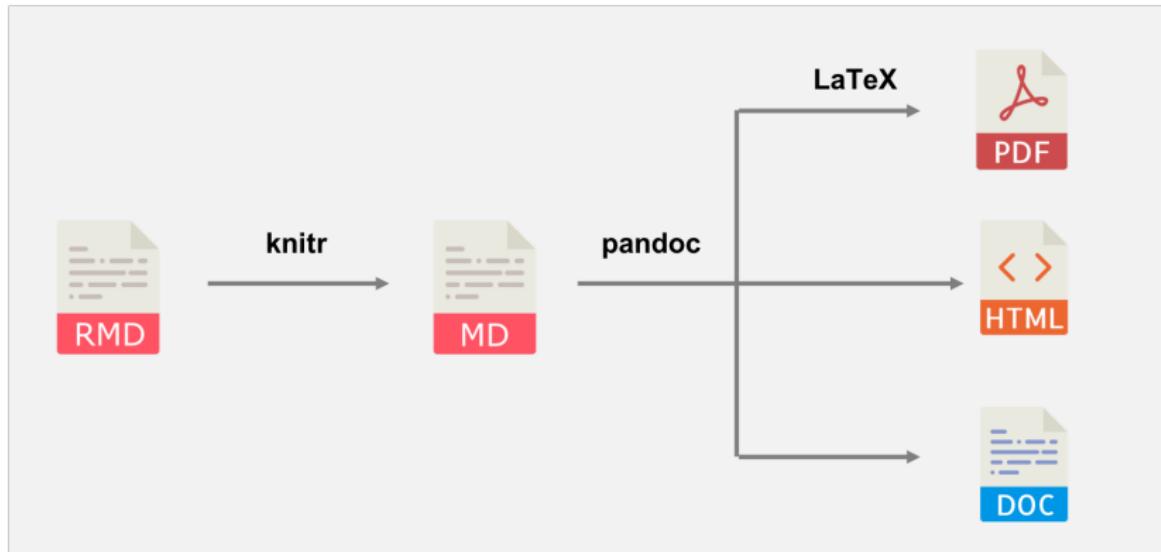
Aquí puedes ver los detalles técnicos de *TinyTeX* y aquí los FAQ.

Rmarkdown

TEXT. CODE. OUTPUT.
(GET IT TOGETHER, PEOPLE.)



¿Como funciona?



Anatomía de RMarkdown

► YAML Header

```
---
```

```
title: "RProjects y RMarkdown"
author: Alejandra G.Cabanillas
output: html_document
```

```
---
```

- ▶ Va al principio del documento.
- ▶ Lo lee *Pandoc*, *rmarkdown* y *knitr*.
- ▶ Describe que tipo de documento estas generando y es donde se realizan las modificacion de formato y *typesetting*.

Para mas información ver este video, este mini-curso o esta sección

Ventajas

- ▶ Al integrar la creación de figuras con el análisis de datos puedes actualizar documentos de manera casi automática.
- ▶ Facilita modificar figuras durante el proceso de edición.
- ▶ Ahorra tiempo al hacer editar el formato deseado automáticamente (evita mover todo de sitio cuando cambias una imagen)

Integracion con *Rstudio Projects*

- ▶ Lo único que debes hacer es guardar tu documento de markdown (**.Rmd**) en el directorio de el *Project*
- ▶ Cualquier imagen o datos que vayan a ser utilizados en el documento también deben estar en el mismo directorio.

Ejemplo de organizacion en un *Project*

The screenshot shows a file browser window titled "test_project". The interface includes a toolbar with various icons for file operations like copy, paste, and search. Below the toolbar is a header bar with a folder icon and the project name. The main area is a table listing files and folders:

Name	Date Modified	Size	Kind
▼ data	Today at 1:30 PM	6 KB	Folder
► processed	Today at 1:30 PM	Zero bytes	Folder
► raw	Today at 1:30 PM	Zero bytes	Folder
► output	Today at 1:30 PM	Zero bytes	Folder
► scripts	Today at 1:30 PM	Zero bytes	Folder
► temp	Today at 1:33 PM	Zero bytes	Folder
R test_project.Rproj	Today at 1:52 PM	205 bytes	R Project

Mas detalles aqui.

Diferencias entre un *Script* y un documento de *Markdown*

- ▶ En un *script* todo es código que va a ser ejecutado (menos los comentarios)
- ▶ En *R Markdown* solo los *chunks* son ejecutables.
- ▶ Un archivo de *R Markdown* es básicamente texto con secciones ejecutables.
- ▶ Piensa en ello como un procesador de texto muy flexible con **R** integrado.

rmarkdown :: CHEAT SHEET

What is rmarkdown?

- .Rmd files** - Develop your code and ideas side-by-side in a single document. Run code as individual chunks or all at once.
- Dynamic Documents** - Write together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS Powerpoint.
- Reproducible Research** - Upload link to, or attach your .Rmd file to share. Anyone can run or fork your code to reproduce your work.

Workflow

- Open a new **.Rmd file** in the RStudio IDE by going to **File > New File > R Markdown**.
- Embed code** in chunks. Run code by line, by chunk, or all at once.
- Write text** and add tables, figures, images, and citations. Format with Markdown syntax or the RStudio Visual Markdown Editor.
- Set output format(s) and options** in the YAML header. Customize themes or add parameters to execute or add interactivity with shiny.
- Save and render** the whole document. Knit periodically to preview your work as you write.
- Share your work**:

The screenshot illustrates the RStudio interface for working with rmarkdown. The Source Editor on the left displays R code and Markdown text, with tools for saving, rendering, and sharing. The Rendered Output pane on the right shows the final HTML output, including a title and a section titled 'R Markdown' with its rendered content.

Embed Code with knitr

CODE CHUNKS

Surround code chunks with `` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after it.

```
```{r chunk-label, include=FALSE}
summary(mtcars)
...```

```

### SET GLOBAL OPTIONS

Set options for the entire document in the first chunk.

```
```{r include=FALSE}
knitr::opts_chunk$set(message = FALSE)
```

INLINE CODE

Insert `r <code>` into text sections. Code is evaluated at render and results appear as text.

"Built with r getReversion()"-->"Built with 4.1.0"

OPTION	DEFAULT EFFECTS
echo	TRUE display code in output document
error	TRUE display error messages in doc
eval	FALSE (stop render when error occurs)
include	TRUE include chunk in doc after running
message	TRUE display message messages in document
warning	TRUE display warning messages in document
results	"markup" ("asis" (pass through results)) "hold" (put all results below all code)
fig.align	"default" (left)
fig.alt	NULL alt text for a figure
fig.cap	NULL figure caption as a character string
fig.path	NULL path for generating figure file paths
fig.width & fig.height	7 pixels by 3.5 inches
out.width	rescales output width, e.g. "75%", "300px"
collapse	collapse all sections into a single block
comment	"#>" (add line of results)
child	file(s) to knit and then include
purl	include or exclude a code chunk when extracting source code with knitr::purl()

See more options and defaults by running `str(knitr::opts_chunk$get())`

Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

BUILD YOUR BIBLIOGRAPHY

- Add BibTeX or CSL bibliographies to the YAML header. title: "My Document" bibliography: references.bib link-citations: TRUE

If Zotero is installed locally, your main library will automatically be available.

- Add citations by DOI by searching "from DOI" in the **Insert Citation** dialog.

INSERT CITATIONS

- Access the **Insert Citations** dialog in the Visual Editor by clicking the @ symbol in the toolbar or by clicking **Insert > Citation**.
- Add citations with markdown syntax by typing **@cite** or @cite[]

Insert Tables

Output data frames as tables using `kable(data, caption)`.

```
```{r}
data <- faithful[1:4,]
knitr::kable(data,
 caption = "Table with kable()")
...```

```

Other table packages include `flextable`, `gt`, and `kableExtra`.

## Write with Markdown



The syntax on the left renders as the output on the right.

Plain text.

End a line with two spaces to start a new paragraph.

Also end with a backslash to make a new line.

\*italics\* and \*\*bold\*\*

superscript<sup>23</sup>/subscript<sub>24</sub>

~strikethrough<sub>25</sub>

escaped: ``\`\\`

endash --, emdash ---

### Header 1

### Header 2

### Header 3

### Header 4

### Header 5

### Header 6

- unordered list
  - Item 1
  - Item 2

- ordered list
  - Item 1
  - Item 2

<http://www.rstudio.com/>

This is a link.

This is another link.

At the end of the document:

- [id] url
- [Caption] image.png
- or [Caption][id]

At the beginning of the document:

- [id] image.png

verbatim code

multiple lines of verbatim code

block quotes

$$\text{equation: } S^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2$$

$$\text{equation block: } S^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2$$

horizontal rule:

---

Right	Left	Default	Center
12	12	12	12
12	12	12	12
123	123	123	123
1	1	1	1

### HTML Tables

- # Results (dataset)
- # Plots (test)

## Tables

more test





## Set Output Formats and their Options in YAML

Use the document's YAML header to set an output format and customize it with `output` options.

```
title: "My Document"
author: "Author Name"
output:
 html_document:
 toc: TRUE
```

**Indent format 2 characters, indent options 4 characters**

OUTPUT FORMAT	CREATES
html_document	.html
pdf_document*	.pdf
word_document	Microsoft Word (.docx)
powerpoint_presentation	Microsoft Powerpoint (.pptx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Markdown for Github
ioslides_presentation	ioslides HTML slides
sldy_presentation	Silky HTML slides
beamer_presentation*	Beamer slides

\* Requires LaTeX, use `tinytex::install_tinytex()`  
Also see `flexdashboard`, `bookdown`, `downlist`, and `blogdown`.

IMPORTANT OPTIONS	DESCRIPTION	HTML	PDF	MS Word	MS PowerPoint
anchor_sections	Show section anchors on mouse hover (TRUE or FALSE)	X			
citation_package	The LaTeX package to process citations ("default", "natbib", "biblatex")	X			
code_download	Give readers an option to download the Rmd source code (TRUE or FALSE)	X			
code_folding	Let readers to toggle the display of R code ("none", "hide", or "show")	X			
css	CSS or SCSS file to use to style documents (e.g. "style.css")	X			
dev	Graphics device to use for figure output (e.g. "png", "pdf")	X	X		
df_print	Method for printing data frames ("default", "kable", "tibble", "paged")	X	X X X		
fig_caption	Should figures be rendered with captions (TRUE or FALSE)	X	X X X		
highlight	Syntax highlighting ("tango", "pygments", "kate", "zenburn", "textmate")	X	X X		
includes	File of content to place in doc ("in_header", "before_body", "after_body")	X	X X		
keep_md	Keep the Markdown .md file generated by knitting (TRUE or FALSE)	X	X X X		
keep_tex	Keep the intermediate TEX file used to convert to PDF (TRUE or FALSE)	X	X X X		
latex_engine	LaTeX engine for producing PDF output ("pdflatex", "xelatex", or "luatex")	X	X		
reference_docx_dot	docs/opts file containing styles to copy in the output (e.g. "file.docx", "file.pdfx")	X	X X		
theme	Theme options (see Bootswatch and Custom Themes below)	X			
toc	Add a table of contents at start of document (TRUE or FALSE)	X	X X X		
toc_depth	The lowest level of heading to add to table of contents (e.g. 2, 3)	X	X X X		
toc_float	Float the table of contents to the left of the main document content (TRUE or FALSE)	X	X		

Use `?output_formats` to see all of a format's options, e.g. `?html_document`

## More Header Options

### PARAMETERS

Parameterize your documents to reuse with new inputs (e.g., data, values, etc.).

1. Add parameters in the header as sub-values of params.

```
params:
 state: "hawaii"
```

```
 ... (r)
 data <- df[, params$state]
```

```
 summary(data)
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

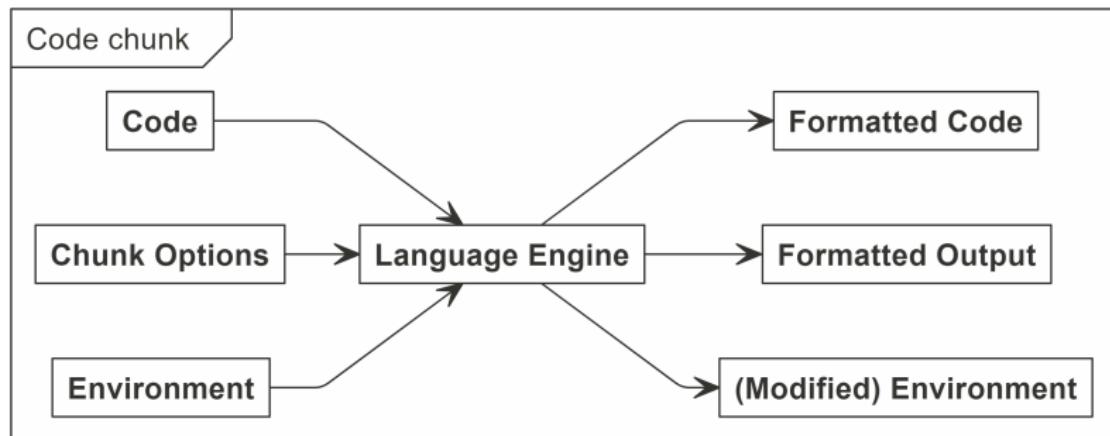
```
 ... (r)
 echo: TRUE
```

```
 ... (r)
 echo: FALSE
```

```
 ... (r)
 echo: TRUE
```

# Integrando código de R en tu documento

- ▶ Crean nuestros reportes dinámicos y mantienen el documento sincronizado.
- ▶ Funciona así:



## Como incluir tu código:

- ▶ *Chunk:*

```
x <- 5 # radius of a circle
```

- ▶ *In-line*

Para un círculo con radio 5, su área es 78.5398163.

- ▶ En la línea anterior los valores han sido calculados directamente desde las variables asignadas. En el documento se ve así :

```
Para un círculo con radio `r x`, su área es `r pi * x^2`
```

## Opciones disponibles

- ▶ **include = FALSE** Ejecuta el código pero no lo enseña en el documento. *R Markdown* puede utilizar ese *output* en el resto del documento.
- ▶ **echo = FALSE** Evita que el código se ejecute. Pero enseña los resultados. Es una manera muy práctica de incluir gráficos.
- ▶ **message = FALSE** Evita que los mensajes producidos por el código se muestren en el documento. Por ejemplo cuando instalamos paquetes o una gráfica no está del todo ajustada.
- ▶ **warning = FALSE** Evita que se muestren los mensajes de error en el documento final. Es muy completamente cuando solo quieres enseñar un trozo de código complejo y no ejecutarlo completamente.

Hay más de 50 opciones que pueden modificar los *chunks* dependiendo de tus necesidades.

Lista completa

## Integrando imágenes

- ▶ Puedes generarlas dentro de el documento.
- ▶ También puedes importarlas de otras plataformas.

Informacion basica sobre figuras en R Markdown y aqui consejos y trucos

## Generando figuras

- ▶ Escribe el código necesario para generala dentro de un *chunk*
- ▶ Utiliza las opciones de los *chunks* para modificar el tamaño y agregar leyendas.

Aqui para un ejemplo en detalle, también esta en mas detalle en el seminario 2.

## Añadir una leyenda

- ▶ Simplemente vamos a agregar  $fig.cap = "tu leyenda"$  en las opciones del *chunk*

```
plot(mpg ~ hp, mtcars)
```

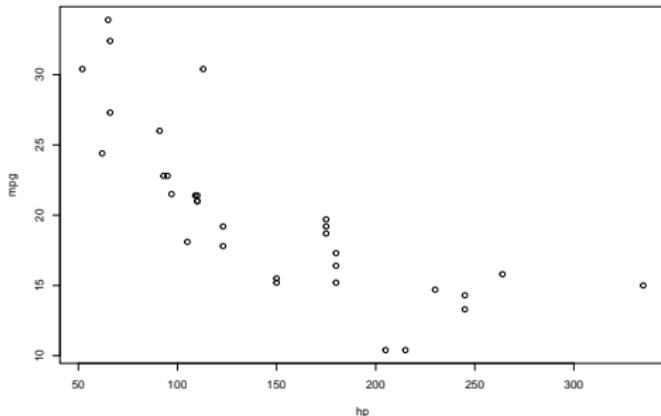
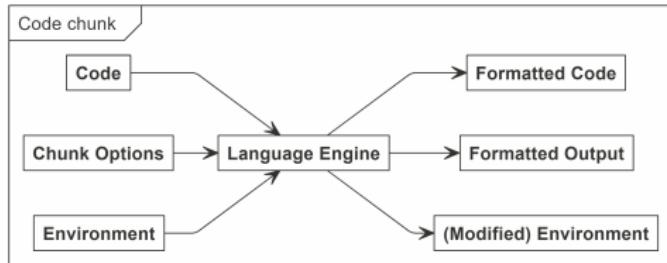


Figure 1: Un ejemplo en R base

## Añadir una imagen ya guardada

- ▶ Especifica donde esta.
- ▶ Que formato es.
- ▶ El tamaño deseado.
- ▶ También puedes utilizar **fig.cap**

```
```{r, echo = FALSE, out.width="100%"}  
knitr::include_graphics("./images/chunks.png")  
```
```



## Que pasa si quieres aplicar un *script* externo?

- ▶ Se utiliza de la misma manera que en un archivo .R

```
source("your-script.R", local = knitr::knit_global())
or sys.source("your-script.R", envir = knitr::knit_globa
```

## Programmatic reporting

- ▶ Cuando todo lo anterior es aplicado dentro de un documento tenemos lo que se conoce como *Programmatic reporting*.
- ▶ Un documento que se actualiza fácilmente cuando cambian los datos que se han utilizado para generarlo.

## R Markdown script

```
1<--> #!/usr/bin/env Rscript
2 title: "Outbreak Situation Report"
3 date: "4/24/2021"
4 output: word_document
5
6
7 ## In setup, echo=FALSE
8 pacman::p_load(dplyr, here, tidyverse, janitor, incidence2, flextable)
9 timelist <- rio::import(here("data", "case_linelists", "timelist_cleaned.rds"))
10
11
12 This report is for the Incident Command team of the fictional outbreak of Ebola cases.
13 ## As r format(max(timelist$date_hospitalized, na.rm=T), "Nd Nd") there have
14 been r nrow(timelist) cases reported as hospitalized.
15
16 ## Summary table of cases by hospital
17
18 ## {r, echo=F, out.height="715px"}
19 timelist %>%
20 group_by(hospital) %>%
21 summarise(cases = n(),
22 deaths = sum(outcome == "Death", na.rm=T),
23 recovered = sum(outcome == "Recovered", na.rm=T)) %>%
24 adorn_totals() %>%
25 gflextable()
26
27
28 ## Epidemic curve by age
29 ## {r, echo=F, warning=F, message=F, out.height = "750px", out.width="100%"}
30 # create epidemic
31 age_outbreak <- incidence(
32 timelist,
33 date_index = date_onset, # date of onset for x-axis
34 interval = "week",
35 groups = age_cat)
36
37 # plot
38 plot(age_outbreak, n_breaks = 3, fill = age_cat, col_pal = muted, title = "Epidemic
39 curve by age group")
40
```

YAML sets title, date, and output type

Code chunk loads packages and data

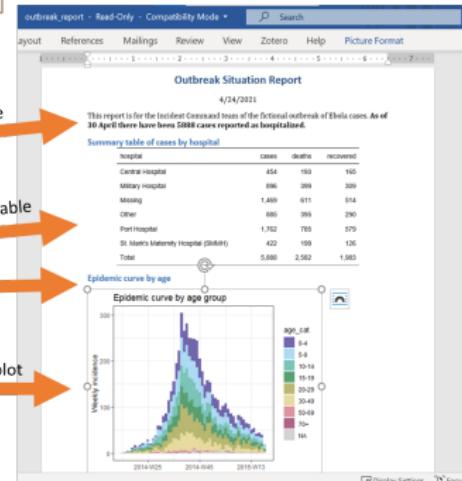
Text and in-line code

Code chunk makes table

## Headings

Code chunk makes plot

## Output (e.g. Word document)



## Source

## Todo lo que ya habeis aprendido en version CV

- ▶ Programming Language: R
- ▶ Exploratory Data Analysis tools: tidyverse
- ▶ Dataviz tools: ggplot2
- ▶ Programmatic reporting: RMarkdown

Source

## Referencias

- ▶ Reports en R Markdown
- ▶ R Markdown for Scientists. Empieza aqui, tiene las explicaciones breves y ejercicios.
- ▶ R Markdown Cookbook para todas las preguntas que se te puedan ocurrir.
- ▶ R Markdown: The Definitive Guide