

# R Markdown :: GUÍA RÁPIDA



## ¿Qué es R Markdown?



**.Rmd** • Un archivo R Markdown (.Rmd) es un registro de tu investigación. Contiene el código que un científico necesita para reproducir el trabajo junto con la narración que el lector requiere para comprenderlo.

**Investigación reproducible** • Con el click de un botón, o el tipeo de un comando, puedes volver a ejecutar el código en un archivo R Markdown para reproducir tu trabajo y exportar los resultados como un informe terminado.

**Documentos dinámicos** • Puedes optar por exportar el informe terminado en una variedad de formatos, incluidos documentos html, pdf, MS Word o RTF; diapositivas, cuadernos y más basados en html o pdf.

**R Markdown**

**RStudio**

**R Markdown**

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
summary(cars)
```

##	speed	dist
## Min.	: 4.0	Min. : 2.00
## 1st Qu.	:12.0	1st Qu.: 26.00
## Median	:15.0	Median : 36.00
## Mean	:15.4	Mean : 42.98
## 3rd Qu.	:19.0	3rd Qu.: 56.00
## Max.	:25.0	Max. :120.00

For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

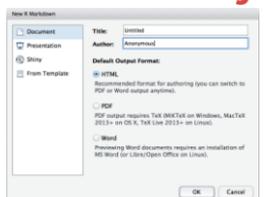
## Estructura

**Encabezado YAML**  
Sección opcional del render (ej. pandoc)  
Opciones escritas como pares clave-valor (YAML).  
Al comienzo del archivo  
Entre líneas de ---

**Texto**  
Narración formateada con markdown, mezclada con:

**Bloques de código**  
Fragmentos de código incluido. Cada bloque:  
Comienza con `{r}`  
Termina con `}`  
R Markdown ejecuta el código y agrega los resultados al documento.  
Utiliza la ubicación del archivo .Rmd como el **directorio de trabajo**.

## Flujo de trabajo



- 1 **Abre un nuevo archivo .Rmd** en File ► New File ► R Markdown. Utiliza el asistente que se abre para rellenar previamente el archivo con una plantilla.
- 2 **Escribe el documento** editando la plantilla.
- 3 **Compila (Knit) el documento para crear un reporte**; usa el botón *knit* o `render()` para hacerlo.
- 4 **Vista previa del resultado** en ventana de IDE
- 5 **Publica** (opcional) al servidor web.
- 6 **Examina el registro de construcción** en la consola R Markdown
- 7 **Usa el archivo de salida** que se guarda junto a .rmd.

## Parámetros

Parametriza tus documentos para reutilizarlos con nuevos inputs (ej. datos, valores, etc.)

```
---
params:
  n: 100
  d: !r Sys.Date()
---
```

1. **Agrega parámetros** • Establece parámetros en el encabezado como subvalores de parámetros.
2. **Llama parámetros** • Llama a los parámetros en el código como `params$<nombre>`
3. **Establece parámetros** • con *knit with parameters* o con el argumento `params del render()`:  
`render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01")))`

## Renderizar

Usa `rmarkdown::render()` para renderizar/*knit* desde línea de comando. Argumentos importantes:

<b>input</b> - archivo a compilar	<b>output_options</b> - lista de opciones de renderizado (como en YAML)	<b>output_file</b> - nombre del archivo de salida	<b>output_dir</b> - directorio de salida	<b>params</b> - lista de parámetros a usar	<b>envir</b> - entorno en el cual evaluar bloques de código	<b>encoding</b> - formato de codificación del archivo a compilar (input)
-----------------------------------	-------------------------------------------------------------------------	---------------------------------------------------	------------------------------------------	--------------------------------------------	-------------------------------------------------------------	--------------------------------------------------------------------------

## Insertar código con *knitr syntax*

**CÓDIGO INCRUSTADO EN TEXTO**  
Insertarlo con ``r <code>``. Los resultados aparecen en el texto sin el código

Built with ``r getRversion()`` → Built with 3.2.3

**BLOQUES DE CÓDIGO**  
Una ó más líneas rodeadas con `{r}` and `}`. Ubica las opciones del bloque entre llaves luego de `r`. Insert with `{r}`

```
{r}
getRversion()
## [1] '3.2.3'
```

**OPCIONES GLOBALES**  
Definir con `knitr::opts_chunk$set()`, ej.

```
{rinclude=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## OPCIONES IMPORTANTES DE LOS BLOQUES DE CÓDIGO

- |                                                                                                     |                                                                                                                                 |                                                                                  |                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cache</b> - resultados almacenados en caché para futuras compilaciones / knits (default = FALSE) | <b>dependson</b> - dependencias de los bloques al caché (default = NULL)                                                        | <b>fig.align</b> - 'left', 'right', o 'center' (default = 'default')             | <b>message</b> - mostrar mensajes del código en el documento (default = TRUE)                                                                                                       |
| <b>cache.path</b> - directorio en donde guardar los resultados en caché (default = "cache/")        | <b>echo</b> - mostrar el código en el documento de output (default = TRUE)                                                      | <b>fig.cap</b> - título del gráfico como cadena de caracteres (default = NULL)   | <b>results</b> (default = 'markup')<br>'asis' - incluir los resultados en el output<br>'hide' - no mostrar resultados<br>'hold' - incluir los resultados al final de todo el código |
| <b>child</b> - archivos para compilar y luego incluir (default = NULL)                              | <b>engine</b> - lenguaje de programación utilizado en el bloque de código (default = 'R')                                       | <b>fig.height, fig.width</b> - dimensiones de los gráficos en pulgadas.          | <b>tidy</b> - emproljar el código para mostrar (default = FALSE)                                                                                                                    |
| <b>collapse</b> - combinar todos los outputs en un único bloque (default = FALSE)                   | <b>error</b> - mostrar mensaje de error en el doc (TRUE) o detener la compilación cuando hay un error (FALSE) (default = FALSE) | <b>highlight</b> - resaltar código fuente (default = TRUE)                       | <b>warning</b> - mostrar las advertencias del código en el documento (default = TRUE)                                                                                               |
| <b>comment</b> - prefijo para cada línea de resultados (default = '##')                             | <b>eval</b> - ejecutar el código dentro del bloque (default = TRUE)                                                             | <b>include</b> - incluir bloques en el doc luego de ejecutarlos (default = TRUE) |                                                                                                                                                                                     |

Options no mencionadas: `R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts`

## Documentos interactivos

Convierte tu reporte en un documento interactivo de Shiny en 4 pasos:

1. Añade `runtime:shiny` al encabezado YAML.
2. Utiliza funciones `input` de Shiny para agregar objetos de input.
3. Utiliza funciones `render` de Shiny para incluir salidas reactivas.
4. Renderiza con `rmarkdown::run` o clickeando Run Document en el IDE de RStudio

```
---
output: html_document
runtime: shiny
---
{r, echo = FALSE}
numericInput("n", "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```

How many cars?  
5

	speed	dist
1	4.00	2.00
2	4.00	10.00
3	7.00	4.00
4	7.00	22.00
5	8.00	16.00

Incorpora una app completa a tu documento con `shiny::shinyAppDir()`

**Publica en RStudio Connect**, para compartir documentos R Markdown de forma segura, agendar actualizaciones automáticas e interactuar con parámetros en tiempo real [www.rstudio.com/products/connect/](http://www.rstudio.com/products/connect/)





# Pandoc's Markdown

Escriba con la sintaxis de la izquierda para crear el efecto de la derecha (luego del renderizado)

Texto plano Termine la línea con dos espacios para empezar un nuevo párrafo.  
 Texto plano Termine la línea con dos espacios para empezar un nuevo párrafo.  
 \*ítilica\* y \*\*negrita\*\*  
 ` código verbatim`  
 sub/superíndice^2^~2~  
 ~tachado~  
 Escapado: \\*\_\|\  
 guión en: --, guión emm: ---  
 ecuación:  $A = \pi * r^2$   
 bloque de ecuación:  

$$E = mc^2$$

Texto plano Termine la línea con dos espacios para empezar un nuevo párrafo.  
**ítilica** y **negrita**  
 código verbatim  
 sub/superíndice<sup>2</sup>  
 tachado  
 escapado: \* \_ \|\  
 guión en: --, guión emm: ---  
 ecuación:  $A = \pi * r^2$   
 bloque de ecuación:  

$$E = mc^2$$

> Bloque de cita.  
autor

Bloque de cita. autor

# Encabezado 1 {#anchor}

## Encabezado 2 {#css\_id}

### Encabezado 3 {.css\_class}

#### Encabezado 4

##### Encabezado 5

##### Encabezado 6

<!--Comentario de texto-->

\textbf{Tex se ignora en HTML}

<em>HTML se ignora en pdfs</em>

<http://www.rstudio.com>

[link](www.rstudio.com)

lr a [Encabezado 1]{#anchor}

Imagen:

![Título aquí](rmd.png)

\* Lista sin orden  
+ sub-ítem 1  
+ sub-ítem 2  
- sub-sub-ítem 1

\* ítem 2

continuar (indentar 4 espacios)

1. Lista ordenada  
2. ítem 2  
i) sub-ítem 1  
A. sub-sub-ítem 1

(@) una lista cuya numeración continúa

(@) luego de una interrupción

: Definición 1

continúa

Definición 1

# Establecer opciones de renderizado con YAML

Cuando renderiza, R Markdown

1. ejecuta el código de R, e incrusta resultados y texto en un archivo .md con knitr
2. Luego convierte el archivo .md en el formato finalizado con pandoc



Establezca el formato del documento resultante en el encabezado YAML

```
---
output: html_document
# Body
```

el valor

html\_document

pdf\_document

word\_document

odt\_document

rtf\_document

md\_document

github\_document

ioslides\_presentation

slidy\_presentation

beamer\_presentation

Genera

Html

pdf (requiere Tex)

Microsoft Word (.docx)

OpenDocument Text

Formato de texto enriquecido

Markdown

Markdown compatible con Github

Diapositivas ioslides HTML

Diapositivas slidy HTML

Diapositivas Beamer pdf (requiere Tex)

Personalice outputs con sub-opciones (listadas a la derecha):

```
---
output: html_document:
  code_folding: hide
  toc_float: TRUE
# Body
```

Indentar 2 espacios

Indentar 4 espacios

Tabsets html (botones de navegación)

Use la clase css Tablet para ubicar subencabezados en tabs

```
# Tabset {.tabset .tabset-fade .tabset-pills}
## Tab 1
texto 1
## Tab 2
texto 2
### End tabset
```

Tabset

Tab 1

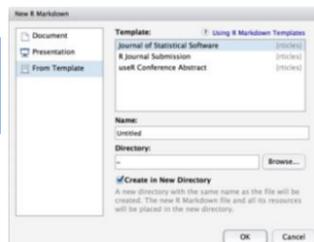
Tab 2

End tabset

# Plantilla Reutilizable

1. Crea un nuevo paquete con un directorio inst/rmarkdown/templates.
2. En el directorio, crea una carpeta que contenga: **template.yaml** (ver más abajo) **skeleton.Rmd** (contenidos de la plantilla) cualquier archivo de apoyo
3. Instala el paquete
4. Accede a la plantilla en el asistente en File ► New File ► R Markdown template.yaml

```
---
name: My Template
---
```



# Tablas

Muchas funciones formatean datos de R a tablas

Table with kable	
eruptions	waiting
3.600	79
1.800	54
3.333	74
2.283	62

eruptionswaiting	
1	3.60 79.00
2	1.80 54.00
3	3.33 74.00
4	2.28 62.00

Table with stargazer	
eruptionswaiting	
1	3.600 79
2	1.800 54
3	3.333 74
4	2.283 62

```
data <- faithful[1:4,]
knitr::kable(data, caption = "Table with kable")
print(xtable::xtable(data, caption = "Table with xtable",
  type = "html", html.table.attributes = "border=0"))
stargazer::stargazer(data, type = "html", title = "Table with stargazer")
```

Mas info en los paquetes stargazer, xtable, and knitr.

# Citas y referencias

Crea citas con archivos .bib, .bibtex, .copac, .enl, .json, .medline, .mods, .ris, .wos, .xml files

1. Establece el archivo de referencias y el archivo de estilo CSL 1.0 (opcional) en el encabezado YAML.

```
---
bibliography: refs.bib
csl: style.csl
---
```

Smith cited [@smith04].  
Smith cited without author [-@smith04].  
@smith04 cited in line.

2. Use llaves para citar en el texto
3. Compile. Las referencias se agregarán al final del documento

Smith cited (Joe Smith 2004).  
Smith cited without author (2004).  
Joe Smith (2004) cited in line.

