



Grupo de Usuarios de R de Málaga

# Investigación Reproducible y RMarkdown

Á. Mora y D. López

26 septiembre 2019, 19 h

# Investigación reproducible

## Reproducible Research - RR

El producto final de un análisis, o de una investigación no es solo un informe o un artículo, es también el **entorno computacional completo** para reproducir los resultados, el código y los datos.

# Aspectos de la investigación reproducible

R integra un conjunto de herramientas que permite que cualquier persona puede **reproducir los resultados bajo las mismas condiciones**. Esto es posible gracias a los documentos dinámicos - *dynamic documents* (*literate programming*, Donald Knuth, 1984).

- **Código:** creación de paquetes, mantenimiento de repositorios de código (GitHub...)
- **Entorno de ejecución:** gestión de versiones tanto de paquetes como del propio R: *packrat*, *Docker*...
- **Resultados de los análisis:** generación de informes, presentaciones, con los resultados, que puedan ser leídos e interpretados por otro investigador.

# Documentos dinámicos con R

Desde R podemos generar documentos dinámicos que integren:

- los datos usados
- explicaciones
- código en R o en otros lenguajes
- resultados

Desde R, usando la herramienta denominada RMarkdown podemos generar informes, presentaciones, libros, webs de alta calidad, siguiendo el paradigma de la *investigación reproducible*.

# RMarkdown

# RMarkdown

Un fichero Rmarkdown puede integrar:

- texto, explicaciones,
- código R como herramienta computacional
- código *L<sup>A</sup>T<sub>E</sub>X* para representar fórmulas matemáticas
- Markdown como lenguaje de marcado para la generación de publicaciones
- hojas de estilo CSS para mejorar las presentaciones
- etc.

# RMarkdown

Un fichero Rmarkdown puede integrar:

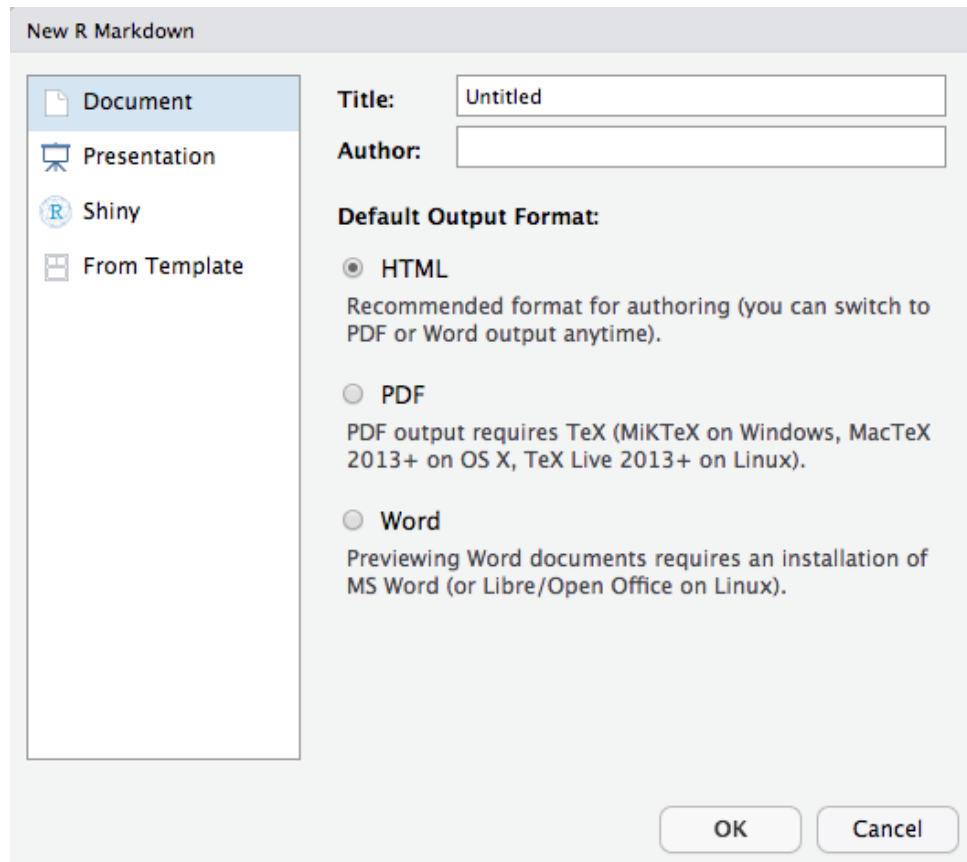
- texto, explicaciones,
- código R como herramienta computacional
- código  $LaTeX$  para representar fórmulas matemáticas
- Markdown como lenguaje de marcado para la generación de publicaciones
- hojas de estilo CSS para mejorar las presentaciones
- etc.

Se necesita el paquete **rmarkdown**, pero no es necesario instalarlo o cargarlo de forma explícita, ya que RStudio hace ambas cosas cuando es necesario.

Se recomienda que se utilice el IDE de **RStudio**, pero no es obligatorio.

# Creación de un documento de RMarkdown

Creamos un nuevo documento a partir del menú **File -> New File -> R Markdown**, que lanzará un asistente que se puede usar para pre-llenar con contenido útil.



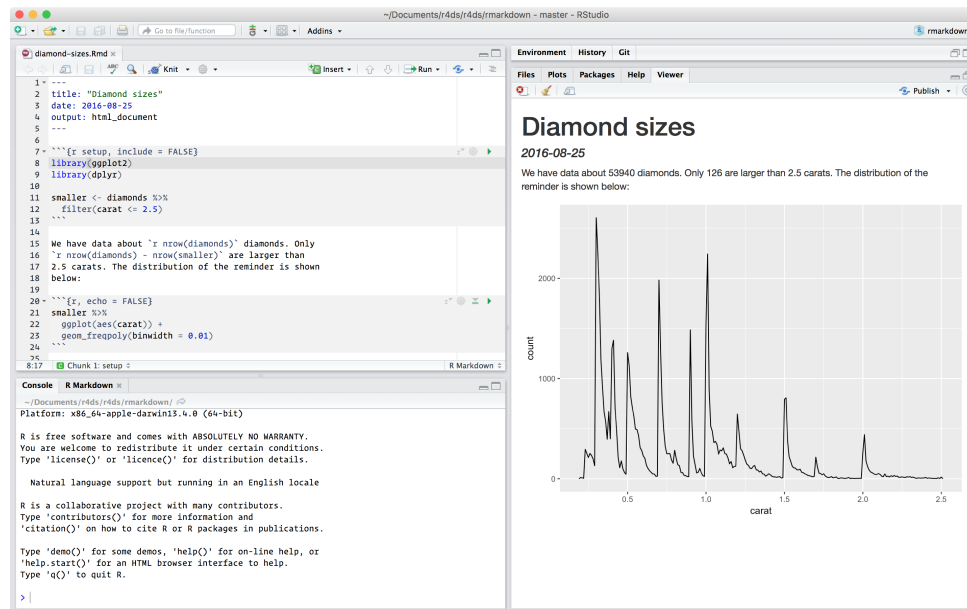


# Análisis de las partes de un RMarkdown

1. **Cabecera:** título y metadatos
2. **Cuerpo del texto:** incluye encabezados, texto normal y trozos de código (*chunks*)

# Compilación del documento

Para producir un informe completo que contenga todo el texto, el código y los resultados, se hace click en el botón "Knit" (en la pequeña barra auxiliar encima del documento), o se puede presionar Cmd/Ctrl + Mayúsc + K. También se puede hacer lo mismo desde la consola, con `rmarkdown::render("nombredelarchivo.Rmd")`. Esto muestra el informe en el panel del visor, creando un archivo (HTML, PDF o *Word*) que se puede compartir.



**¡Vamos a practicar!**

# Documentos

Tipos básicos de documentos (aunque existen más):

- `html_document` genera documentos HTML auto-contenidos.
- `pdf_document` genera un PDF con `\LaTeX` (un sistema abierto de escritura de documentos), que no necesita instalar, ya que RStudio lo hará por defecto si no lo tenía previamente.
- `word_document` para documentos de Microsoft Word (`.docx`).
- `odt_document` para documentos de OpenDocument Text (`.odt`).
- `rtf_document` para documentos en Rich Text Format (`.rtf`).

# Notebooks

Un *notebook*, de tipo `html_notebook`, es una variación de un `html_document`. Las salidas creadas son muy similares, pero el propósito de éste último es distinto.

El objetivo de un `html_document` es comunicar resultados con los *tomadores de decisiones*, mientras que un *notebook* se centra en colaborar con otros científicos de datos.

La diferencia entre ambos propósitos conlleva usos diferentes de la salida, en HTML. Ambos formatos contienen todos los resultados, pero en el *notebook*, el HTML contiene además todo el código fuente.

Esto significa que se puede usar el archivo `.nb.html` generado por el *notebook* de dos formas distintas:

1. Se puede **visualizar en un navegador web**, mostrando todos los resultados. Al contrario de lo que ocurre con el `html_document`, esta visualización siempre incluye una copia incrustada del código fuente que se usó para generarlos.
2. Se puede **editar en RStudio**. Cuando se abre un archivo `.nb.html`, RStudio **recrea de forma automática el archivo `.Rmd`** que lo generó. En un futuro, se prevé que se podrán incluir además archivos auxiliares (por ejemplo, archivos de datos en formato `.csv`), que serían extraídos de forma automática cuando fuera necesario.

# Presentaciones

Se puede usar RMarkdown para producir presentaciones. Se tiene menor control de la apariencia visual que con herramientas como PowerPoint o Keynote, pero la habilidad de incrustar los resultados del código en R dentro de una presentación puede ahorrarnos una gran cantidad de tiempo.

Las presentaciones funcionan dividiendo todo el contenido en *diapositivas*, creando una nueva cada vez que se encuentra en el texto un encabezado de primer (#) o segundo (##) nivel. También se puede insertar una regla horizontal (\*\*\*) para crear una nueva diapositiva sin encabezado.

RMarkdown viene con cuatro formatos de presentación incluidos:

1. `ioslides_presentation` - presentación en HTML ioslides
2. `slidy_presentation` - presentación en HTML con W3C Slidy
3. `beamer_presentation` - presentación en PDF Beamer en  $LATEX$
4. Powerpoint



# Otros paquetes para extender la funcionalidad de RMarkdown

- **vitae** para crear CV de forma dinámica (hay más paquetes que ayudan a recuperar datos de, por ejemplo, ORCID, Google Scholar...)
- **rticles**: Plantillas para la escritura de papers para las revistas y editoriales más relevantes
- **xaringan**: El *ninja* de las presentaciones

# Otros paquetes para extender la funcionalidad de RMarkdown

- **vitae** para crear CV de forma dinámica (hay más paquetes que ayudan a recuperar datos de, por ejemplo, ORCID, Google Scholar...)
- **rticles**: Plantillas para la escritura de papers para las revistas y editoriales más relevantes
- **xaringan**: El *ninja* de las presentaciones
- **blogdown** para la creación de blogs y websites
- **pkgdown** para generar la documentación de un paquete en formato web
- **bookdown** para la creación y publicación de libros

**¡A trabajar!**



# ¡Gracias!

twitter: @dominlopez - @\_RMLg

github: neuroimaginador - RMalagaGroup

Slides created via the R package **xaringan**.