

Plantilla para la realización de informes / memorias

Análisis Exploratorio de Datos. Máster en Ciencia de Datos - UV

Marcelino Martínez Sober

14-10-2022

Índice

1. Documentación <i>online</i>	1
2. Previo. Crea un proyecto.	1
2.1. Salida html	2
2.2. Salida pdf	2
2.2.1. Instalación Tinytex	2
3. Opciones generales chunks	3
4. Librerías cargadas	3
5. ¿Cómo inicio el informe /documento / práctica ?	3
6. Rutas locales	3
7. Cómo incluir figuras.	4
7.1. Subapartado. Opción 1.Figuras en fichero.	4
7.2. Subapartado. Opción 2.Figuras generadas en ejecución.	5
8. Cómo incluir tablas de resultados.	5

Índice de figuras

1. Texto en pie de la figura. Control de tamaño 1	4
2. Texto en pie de la figura. Control de tamaño 2.	4
3. Referencia a figura a partir del nombre del chunk.	5

Índice de cuadros

1. Valores de las variables mostrados en la figura 3.	6
2. Valores de las variables mostrados en la figura 3.	6

1. Documentación *online*

Puedes encontrar un tutorial muy útil sobre **Rmarkdown** en <https://rmarkdown.rstudio.com/lesson-1.html>

2. Previo. Crea un proyecto.

Para cada tarea **crea un nuevo proyecto** con, al menos, una carpeta **./data**, que contenga los datos, (puedes usar carpetas adicionales para organizar la información: programas, **./program**, figuras, **./figure**, etc). El proyecto contendrá al menos un fichero con el código a ejecutar ***.Rmd**.

ATENCIÓN: No uséis rutas con espacios o caracteres extraños.

- `c:\TtoDatos -> SI`
- `c\Ruta de TD Rosa Peña Emoticonos -> NOOOOO!!!`

Esta carpeta es un ejemplo de proyecto:

- **0-Plantilla_RmarkdownPracticas**, es la carpeta de proyecto.
- **0-Plantilla_RmarkdownPracticas.Rproj** es el fichero de proyecto que contiene al menos un programa.
- **Plantilla_Practicas.Rmd** es el programa.
- **data** y **figure** son carpetas locales al proyecto para organizar la información.

Si utilizamos rutas locales, al compartir proyectos basta con comprimir la carpeta del proyecto ya que contendrá todos los ficheros adicionales necesarios.

Al realizar la compilación podemos optar por varios formatos de salida. Los más habituales son **html** y **pdf**, que especificaremos en las opciones *output* de la cabecera del fichero. Es posible obtener varias salidas simultáneamente. El fichero de salida tiene el mismo nombre que el fichero del programa.

2.1. Salida html

Tenemos 2 opciones **html** y **nb.html** (notebook). Son muy similares si bien **nb.html** tiene características adicionales ocultar /visualizar el código (si originalmente estaba visible *echo=TRUE*)

output:

```
# pdf_document: # toc: yes
# toc_depth: 3
# number_sections: yes
# html_notebook:
# echo: yes
# number_sections: yes
# toc: yes
html_document:
echo: yes
number_sections: yes
theme: lumen
toc: yes
```

2.2. Salida pdf

Para generar documentos **pdf** descomentamos las líneas correspondientes.

output:

```
pdf_document: toc: yes
toc_depth: 3
number_sections: yes
# html_notebook:
# echo: yes
```

```
# number_sections: yes
# toc: yes
# html_document:
# echo: yes
# number_sections: yes
# theme: lumen
# toc: yes
```

En el lenguaje *Markdown* (*Rmarkdown* en nuestro caso), es posible incluir elementos de otros lenguajes como *LaTeX*, ampliamente utilizado para la escritura de documentos científicos. Por ejemplo usando *LaTeX* se pueden escribir ecuaciones matemáticas como:

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

Para que la compilación a pdf sea posible es necesario tener instalado en nuestro ordenador un compilador de *LaTeX*. Una alternativa sencilla que nos permite resolver este problema, es instalar un compilador para usarlo solo dentro de R, para ello debes seguir los siguientes pasos:

2.2.1. Instalación **Tinytex**

1. Instala la última versión de R (La puedes descargar desde Cran R)
2. Instala la última versión de RStudio.
3. Desde Rstudio instala la librería *tinytex* para ellos ve escribiendo en la línea de comandos las siguientes instrucciones:

```
■ install.packages("tinytex", dependencies = TRUE)
■ library(tinytex)
■ tinytex::install_tinytex()
```

El proceso puede tardar unos minutos ya que ocupa unos 200MB.

Una vez finalizado el proceso prueba a compilar a *pdf*.

3. Opciones generales chunks

- `include = FALSE` evita que el código y los resultados aparezcan en el archivo terminado. R Markdown aún ejecuta el código en el fragmento y los resultados pueden ser utilizados por otros fragmentos.
- `echo = FALSE` evita que el código, pero no los resultados, aparezcan en el archivo terminado. Esta es una manera útil de incrustar figuras.
- `message = FALSE` evita que los mensajes generados por el código aparezcan en el archivo finalizado.
- `warning = FALSE` evita que las advertencias generadas por el código aparezcan en el informe final.

4. Librerías cargadas

Incluir la lista de librerías necesarias para la ejecución de código en `packages = c("imager", "tidyverse", "knitr")`

```
[1] ".GlobalEnv"      "package:forcats"    "package:stringr"    "package:dplyr"
[5] "package:purrr"    "package:readr"      "package:tidyr"      "package:tibble"
[9] "package:ggplot2"  "package:tidyverse"  "package:kableExtra" "package:imager"
[13] "package:magrittr" "package:knitr"      "package:stats"      "package:graphics"
[17] "package:grDevices" "package:utils"      "package:datasets"   "package:methods"
[21] "Autoloads"        "package:base"
```

5. ¿Cómo inicio el informe /documento / práctica ?

Haz una breve introducción al problema que se desea resolver.

En este documento se describe ... bla bla bla

A continuación ve insertando bloques de código que vayan ilustrando cada una de las etapas así como los resultados obtenidos.

Utiliza diferentes niveles en los títulos para generar automáticamente un índice.

- # Título
- ## Subtítulo
- ### Subsubtítulo

6. Rutas locales

Usa siempre rutas referidas a la carpeta, en la que se encuentra el fichero fuente, y siempre rutas **relativas** a dicha ubicación.

Ejemplos

`ruta1<-'data/tabla1.txt'` : el fichero está en la carpeta datos que cuelga del directorio donde está el código.

`ruta2<- './data/tabla1.txt'` : el fichero está en la carpeta datos que cuelga de un nivel superior

`ruta3<- '../data/tabla1.txt'` : el fichero está en la carpeta datos que cuelga de dos niveles superiores

`ruta4<- '../../data/tabla1.txt'` : el fichero está en la carpeta datos que cuelga de tres niveles superiores

`ruta5<- 'C:/MisDatos/Ej.txt'` **NO USAR NUNCA UNA RUTA ABSOLUTA COMO ESTA**

7. Cómo incluir figuras.

Hay que tener en cuenta que el aspecto final del documento va a depender de si se genera un archivo *html*, o *pdf*, incluso algunas opciones de formato van a ser dependientes del tipo de documento de salida. El documento [Rmarkdown cheat sheet](#)

7.1. Subapartado. Opción 1.Figuras en fichero.

La figura está almacenada en un fichero, en una carpeta local al proyecto.

Podemos controlar las opciones de la figura con parámetros del *chunk*

```
library(imager)
image <- load.image('figure/logoETSE.png')
plot(image, axes = 0)
```

Preferiblemente utiliza imágenes en formato **png** aunque también se aceptan otros formatos.

En el texto podemos referenciar automáticamente las figuras usando `\ref{etiqueta}` con el mismo nombre de etiqueta asociado con `\label{etiqueta}`. Ten en cuenta que no puede haber dos etiquetas iguales.

En **1** tenemos una figura en la que se controla anchura y altura y en **2** se utiliza un porcentaje sobre el tamaño original de la figura.

7.2. Subapartado. Opción 2.Figuras generadas en ejecución.

Dibujamos la gráfica en un *chunk* que nombraremos, por ejemplo. El siguiente se denomina **presFig**



Figura 1: Texto en pie de la figura. Control de tamaño 1



Figura 2: Texto en pie de la figura. Control de tamaño 2.

```
plot(pressure)
```

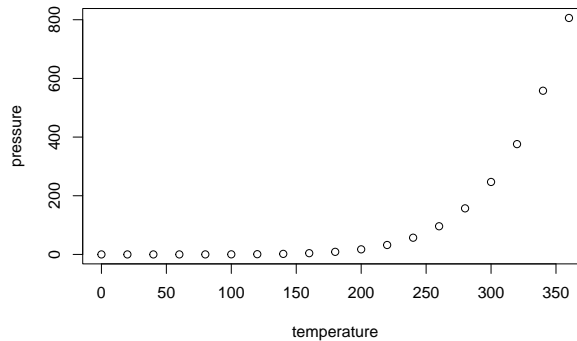


Figura 3: Referencia a figura a partir del nombre del chunk.

Podemos referenciar a la figura usando `\ref{fig::presFig}`.

Por ejemplo diríamos que en la figura 3 se observa que al aumentar la variable `temperature` aumenta la variable `pressure`.

Ten en cuenta que la numeración de las figuras es automática.

8. Cómo incluir tablas de resultados.

También podemos mostrar la información en forma de tabla. La librería `kableExtra` https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html te permite personalizar tus tablas con un sinfín de posibilidades. Tanto en formato *html* como *pdf* https://haozhu233.github.io/kableExtra/awesome_table_in_pdf.pdf

Cuadro 1: Valores de las variables mostrados en la figura 3.

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000
200	17.3000
220	32.1000
240	57.0000
260	96.0000
280	157.0000
300	247.0000
320	376.0000
340	558.0000
360	806.0000

Cuadro 2: Valores de las variables mostrados en la figura 3.

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000
200	17.3000
220	32.1000
240	57.0000
260	96.0000
280	157.0000
300	247.0000
320	376.0000
340	558.0000
360	806.0000