Dynamic documents with R Markdown

Introduction

R Markdown combines report writing/slides/etc and code and data source in one file. The benefits are many and include:

- We can change input data and the document will dynamically update.
- We have a single source for multiple output formats (HTML, PDF, Word).
- We use a single language for multiple output types (presentations, reports, books, papers, ...)

A major challenge for using data in industry, commercial and research settings is that reporting can become separated from the data itself, leading to problems in tracing how a particular graphic, statistic, etc was produced. This harms reproducibility and slows down the updating process, should new data become available.

Getting the packages

We will need (uncomment as necessary)

```
#install.packages("rmarkdown")
```

and possibly the following:

```
#install.packages("tinytex")
#library(rmarkdown)
#tinytex::install_tinytex()
```

The latter is needed for producing pdf files, and creates a local LaTeX installation.

Step-by-step guide

The following steps summarise the basics.

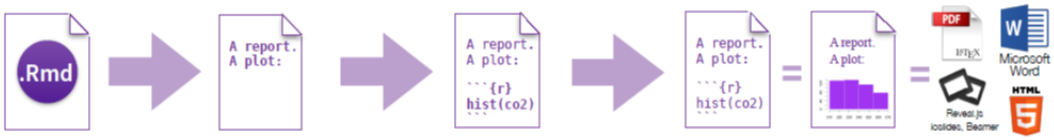
1. Workflow R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

i. Open - Open a file that uses the .Rmd extension.

ii. Write - Write content with the easy to use R Markdown syntax

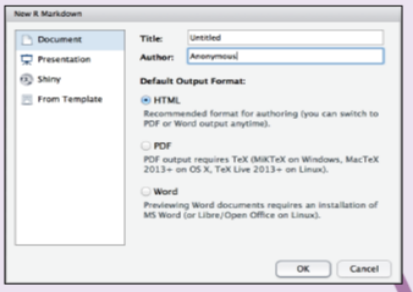
iii. Embed - Embed R code that creates output to include in the report

iv. Render - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.



2. Open File Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File ▸ New File ▸ R Markdown...**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



3. Markdown Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header  | Second Header
------------- | -------------
Table Cell    | Cell 2
Cell 3        | Cell 4
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image: 
horizontal rule (or slide break):

> block quote

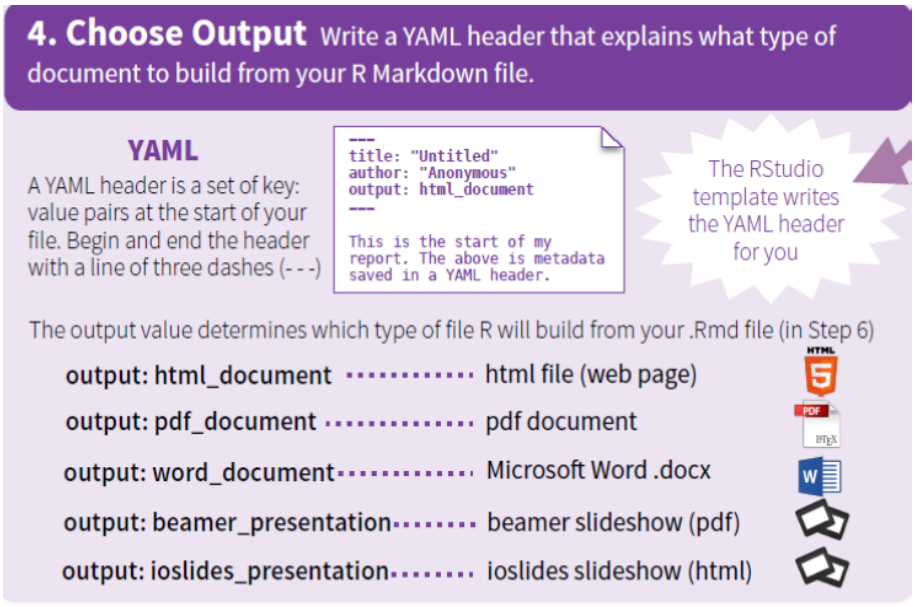
- unordered list
- item 2
    - sub-item 1
    - sub-item 2

1. ordered list
2. item 2
    - sub-item 1
    - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

We also include a **document preamble**, that is a section enclosed between two lines of `---` at the top of the file. This specifies various options for the document, including:

- `title`: the document title
- `author`: (optional) author's name
- `date`: (optional) date for the document
- `output`: for setting options of different output formats (HTML/PDF/Word)



A key consideration is the inclusion of R code and output (chunks). We achieve this via the **triple-backtick fence**, specifying the language to be R.

In the R Markdown file, we use:

````
```{r}
x <- rnorm(10, 3)
mean(x)
```
````

This prints both the code (nicely formatted) and the result of executing that code:

```r
x <- rnorm(10, 3)
mean(x)

## [1] 3.10615
```

This is summarised on the next page with the final render step.

5. Embed Code

Use knitr syntax to embed R code into your report. R will run the code and include the results when you render your report.

inline code

Surround code with back ticks and r.
R replaces inline code with its results.

```
Two plus two equals `r 2 + 2`.
```

▶ Two plus two equals 4.

code chunks

Start a chunk with ```{r}.
End a chunk with ```

```
Here's some code
```{r}
dim(iris)
```
```

▶ Here's some code

```
dim(iris)
```

```
## [1] 150   5
```

display options

Use knitr options to style the output of a chunk.
Place options in brackets above the chunk.

```
Here's some code
```{r eval=FALSE}
dim(iris)
```
```

▶ Here's some code

```
dim(iris)
```

```
Here's some code
```{r echo=FALSE}
dim(iris)
```
```

▶ Here's some code

```
## [1] 150   5
```

option	default	effect
eval	TRUE	Whether to evaluate the code and include its results
echo	TRUE	Whether to display code along with its results
warning	TRUE	Whether to display warnings
error	FALSE	Whether to display errors
message	TRUE	Whether to display messages
tidy	FALSE	Whether to reformat code in a tidy way when displaying it
results	"markup"	"markup", "asis", "hold", or "hide"
cache	FALSE	Whether to cache results for future renders
comment	"##"	Comment character to preface results with
fig.width	7	Width in inches for plots created in chunk
fig.height	7	Height in inches for plots created in chunk

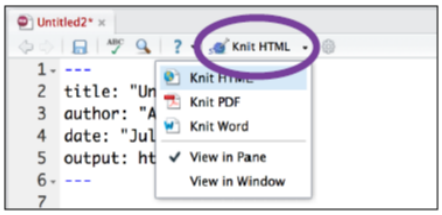
For more details visit yihui.name/knitr/

6. Render

Use your .Rmd file as a blueprint to build a finished report.

Render your report in one of two ways

1. Run rmarkdown::render("<file path>")
2. Click the **knit HTML** button at the top of the RStudio scripts pane



When you render, R will

- execute each embedded code chunk and insert the results into your report
- build a new version of your report in the output file type
- open a preview of the output file in the viewer pane
- save the output file in your working directory

Other steps to the work-flow (not considered here) include the creation of interactive documents.

Key resources

See also:

R Markdown cheat sheet

`https://github.com/rstudio/cheatsheets/raw/main/rmarkdown.pdf`

R Markdown website

`https://rmarkdown.rstudio.com`

R Markdown books

The Definitive Guide `https://bookdown.org/yihui/rmarkdown/`

Cookbook `https://bookdown.org/yihui/rmarkdown-cookbook/`