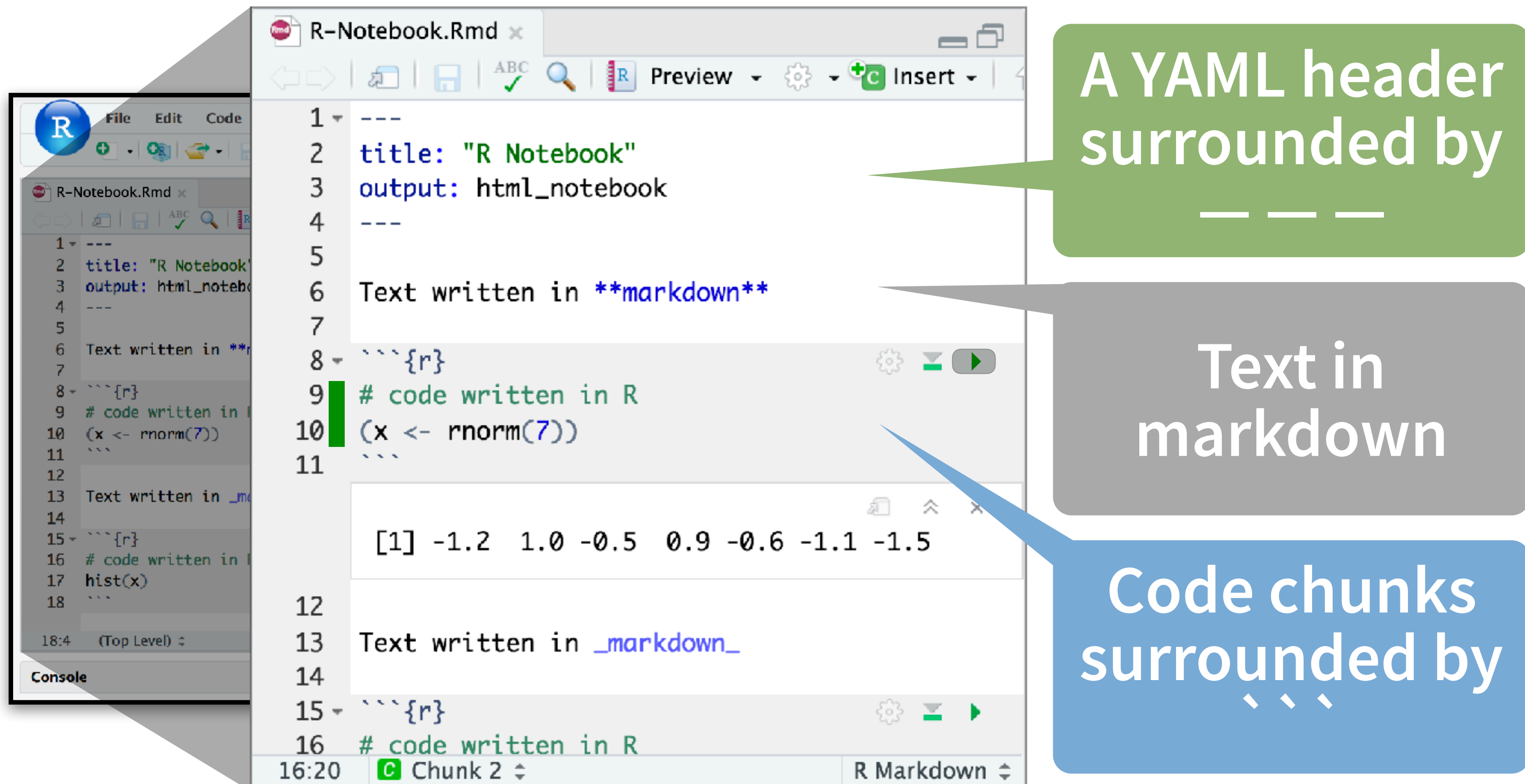


R Markdown



R Markdown

Plain text file with 3 types of content:



The image shows a screenshot of the RStudio interface with an R Markdown file open. The editor displays the following content:

```
1 ---  
2 title: "R Notebook"  
3 output: html_notebook  
4 ---  
5  
6 Text written in markdown  
7  
8 ```{r}  
9 # code written in R  
10 (x <- rnorm(7))  
11 ```  
12  
13 [1] -1.2  1.0 -0.5  0.9 -0.6 -1.1 -1.5  
14  
15 Text written in markdown  
16  
17 ```{r}  
18 # code written in R  
19 hist(x)  
20 ```
```

Callouts explain the three types of content:

- A YAML header surrounded by ---** (points to lines 1-4)
- Text in markdown** (points to line 6)
- Code chunks surrounded by ```** (points to lines 8-11 and 15-18)

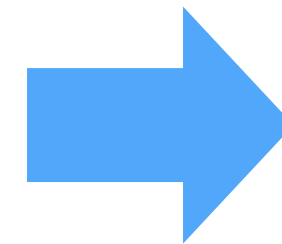
The console at the bottom shows the output of the first code chunk: [1] -1.2 1.0 -0.5 0.9 -0.6 -1.1 -1.5.

Headers

Use # to create headers.

Multiple #'s create lower level headers.

```
# Header 1  
## Header 2  
### Header 3  
#### Header 4  
##### Header 5  
##### Header 6
```



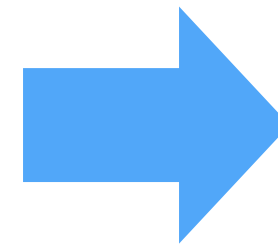
Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

Text

Add two spaces at the end of a line to start a new line

Text is rendered as plain text. Surround text with `_`, `**`, or ``` to format it.

Text○
italics
bold
`code`



Text
italics
bold
`code`

Lists

Use asterisks to make bullet points.

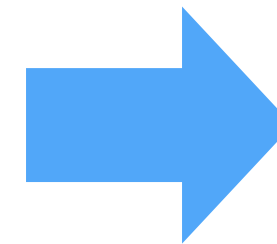
Use numbers to make numbered lists.

Bullets

- * bullet 1
- * bullet 2

Numbered list

1. item 1
2. item 2



Bullets

- bullet 1
- bullet 2

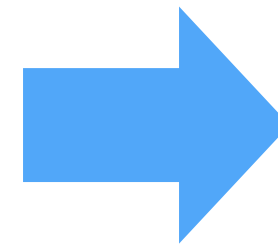
Numbered list

1. item 1
2. item 2

Hyperlinks

Use brackets to denote a link.
Place the URL in parentheses.

This is a
[link](www.git.com).



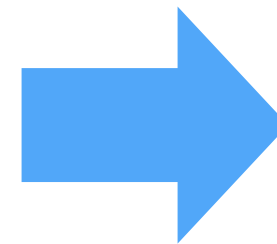
This is a link.

Images

Use a link preceded by an ! to insert an image.

The link text should be a URL (if the image is hosted online), or a file path (if the image is saved as a file)

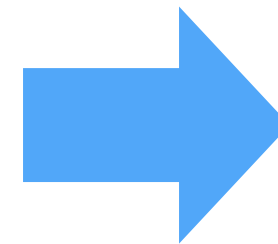
```
  
The RStudio logo.
```



Equations

Write equations with latex math commands and surround them in \$'s.

According to
Einstein,
`$E=mc^{\{2\}}$`



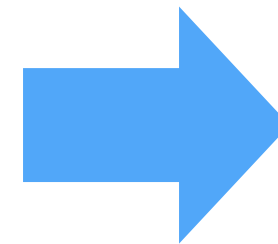
According to
Einstein, $E = mc^2$

Equation blocks

Use two \$'s to make
centered equation blocks.

According to
Einstein,

$$E=mc^2$$



According to
Einstein,

$$E = mc^2$$

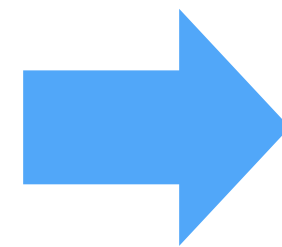
Embed code



Inline code

Place code in a sentence with ``r <code>``. R
Markdown will replace the code with its results.

Today is
``r Sys.Date()``.



Today is 2015-04-16.

Code chunks

Insert a chunk of R code with

```
` `` {r}  
# some code  
` ``
```

When you render the report, R Markdown will run the code and include its results. R Markdown will also remove the `` `` {r}` and `` ```.

Code chunks

Insert a chunk of R code with

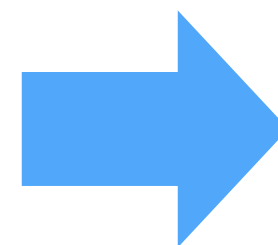
```
` `` {r}  
# some code  
` ``
```



chunk options

By default, R Markdown includes both the code and its results

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



Here's some code

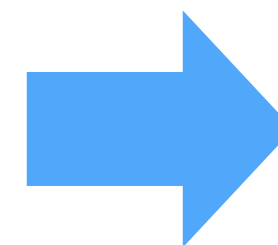
```
dim(iris)
```

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

echo

Add options in the brackets after r.
echo = FALSE hides the code.

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



Here's some code

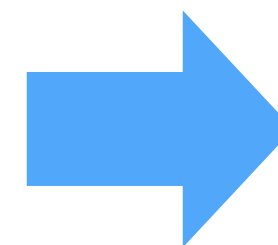
```
## [1] 150 5
```

Very useful
for plots

eval

eval = FALSE prevents the code from being run. As a result, no results will be displayed with the code.

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



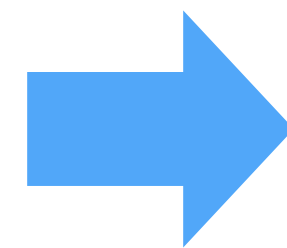
Here's some code

```
dim(iris)
```


include

include = FALSE runs the code, but prevents both the code and the results from appearing (e.g. to setup).

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



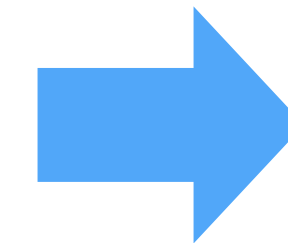
Here's some code

fig.height, fig.width

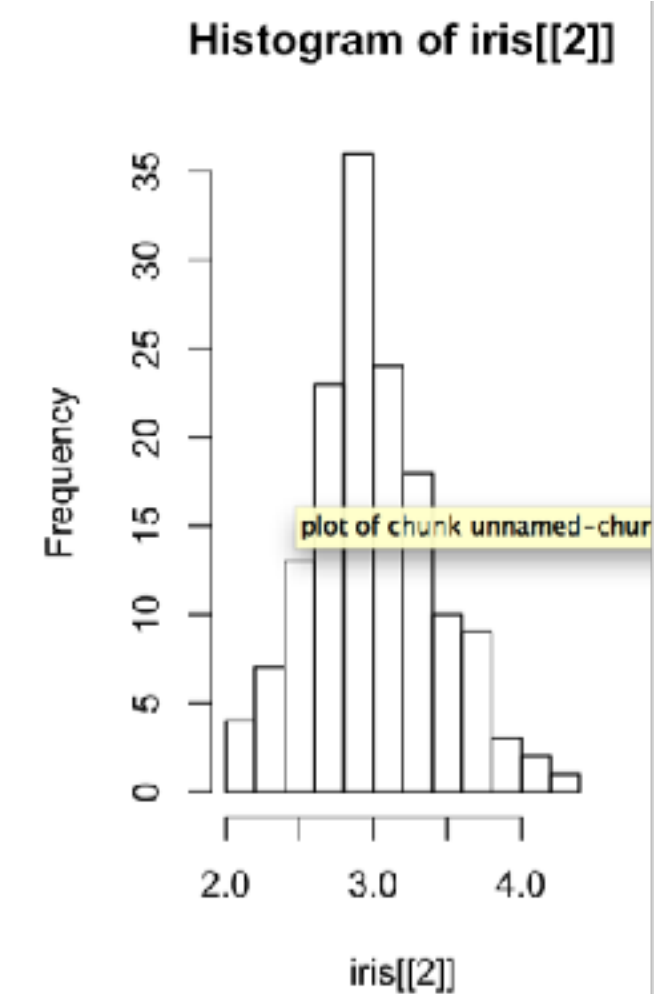
Specify the dimension of plots (in inches) with fig.width and fig.height. Separate multiple arguments with commas.

Here's a plot

```
```{r echo=FALSE, fig.width=3, fig.height=5}  
hist(iris[[2]])
```
```



Here's a plot



Your Turn

Add **include = FALSE** or **echo = FALSE** to each chunk as appropriate to suppress the code.

Set the height of the figure to 2 inches.

Then re-knit your document.

A digital timer with a black border, displaying the time 02:00 in a large, black, digital font. The digits are slightly shadowed, giving it a 3D appearance.

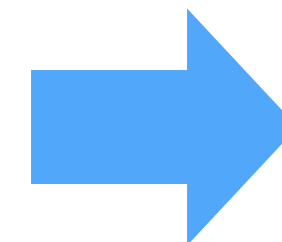
Parameters



YAML

A section of key:value pairs
separated by dashed lines ---

```
---  
title: "Untitled"  
author: "RStudio"  
date: "February 4, 2015"  
output: html_document  
---  
Text of document
```



Untitled

RStudio

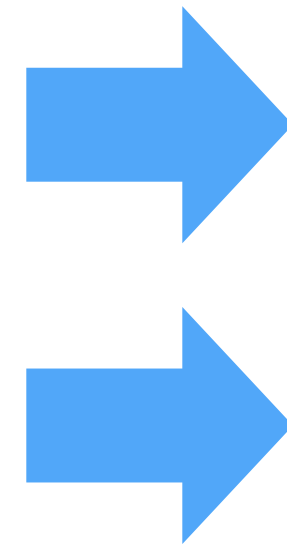
February 4, 2015

Text of document

Parameters

A list of values that you can call in R code chunks

**params list
elements and
values**



```
---  
title: "Untitled"  
output: html_document  
params:  
  filename: "data.csv"  
  symbol: "FB"  
---
```

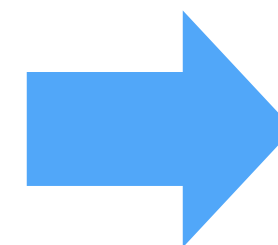
colon

**New line.
Indented two
spaces**

Using Parameters

Call parameter values as elements of the params list, **params\$num**

```
---  
params:  
  num: 42  
---  
The value of the  
parameter is  
`r params$num`, e.g.  
  
`` `{r}  
params$num  
```
```



The value of the  
parameter is 42, e.g.

```
params$num
```

```
[1] 42
```

# render()

Render at the command line with default YAML options

```
library(rmarkdown)
render("03-reports.Rmd")
```

Render at the command line, set parameters.

```
render("03-reports.Rmd",
 params = list(data = "london.csv"))
```



# render() + for

```
datasets <- c("dublin.csv", "london.csv")

for (name in datasets) {
 render("03-reports.Rmd",
 params = list(data = name))
}
```

# How it works



# knitr

# pandoc



ioslides  
slidy, Beamer



Powerpoint



Microsoft Word

# Logistics

1

Knitr runs the document in a fresh R session, which means you need to load the libraries that the document uses *in the document*

# Logistics

1

Knitr runs the document in a fresh R session, which means you need to load the libraries that the document uses *in the document*

2

Objects made in one code chunk will be available to code in later code chunks.

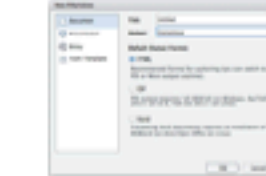


# R Markdown : : CHEAT SHEET

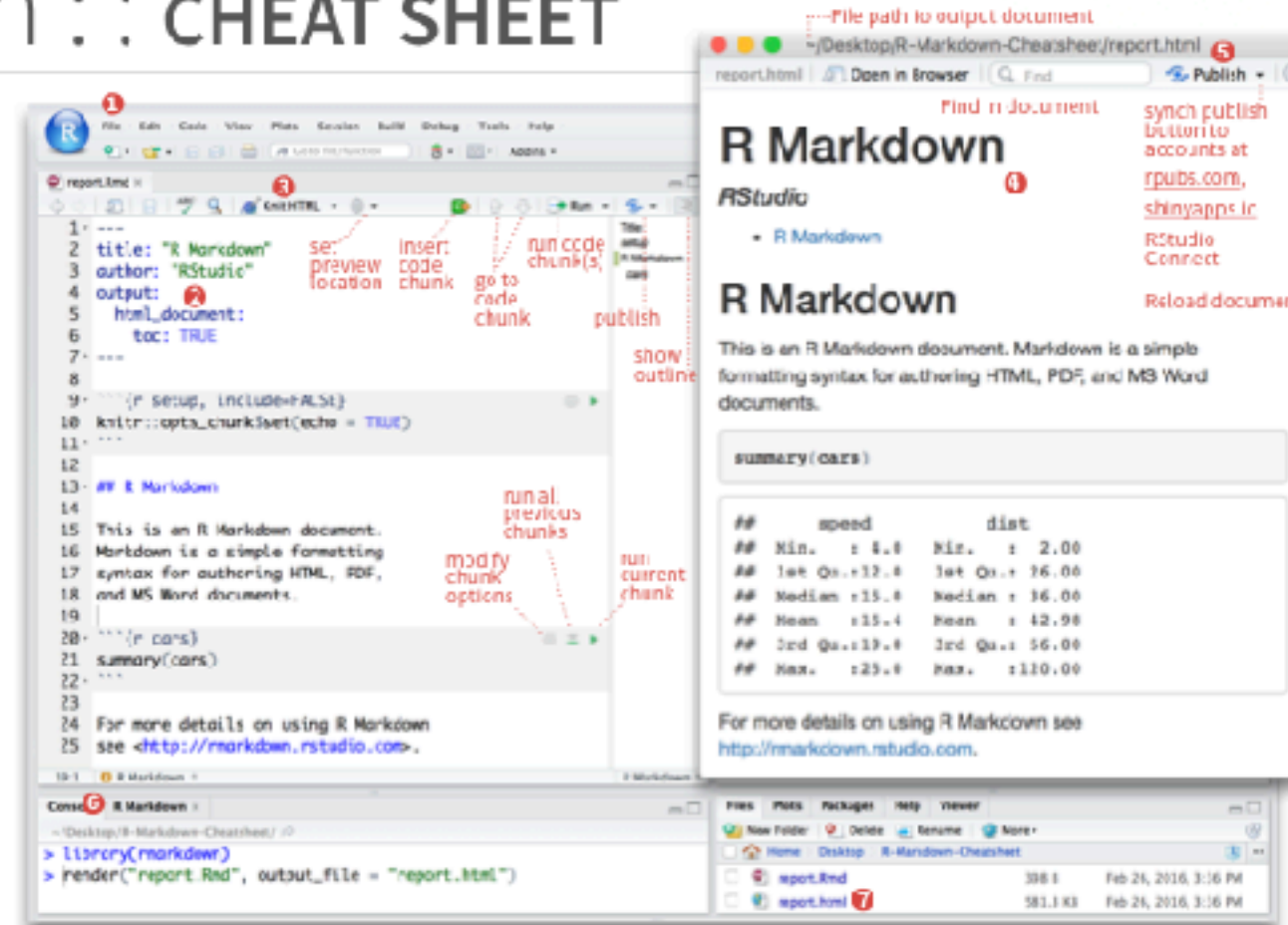
## What is R Markdown?

- Rmd files** - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.
- Reproducible Research** - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.
- Dynamic Documents** - You can choose to export the finished report in a variety of formats, including HTML, PDF, MS Word, or R Markdown documents, HTML or PDF based slides, Notebooks, and more.

## Workflow



1. Open a new Rmd file at File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
2. Write document by editing template.
3. Knit document to create report. Use knit button or render() to knit.
4. Preview Output in DE window.
5. Publish (optional) to web server.
6. Examine build log in R Markdown console.
7. Use output file that is saved alongside Rmd.



## render

Use `render::render()` to render/knit at cmd line. Important args

<b>input</b> - file to render	<b>output_options</b> - list of render options as in YAML	<b>output_file</b> - output file	<b>output_dir</b> - output dir	<b>params</b> - list of params to use	<b>envir</b> - environment to evaluate code chunks in	<b>encoding</b> - of input file
-------------------------------	-----------------------------------------------------------	----------------------------------	--------------------------------	---------------------------------------	-------------------------------------------------------	---------------------------------

## Embed code with knitr syntax

**INLINE CODE**  
Insert with `"r <code>".` Results appear as text, without code.  
Built with `"rgetRversion()"` → Built with `3.2.3`

**CODE CHUNKS**  
One or more lines surrounded with `"{r}"` and `"{r}"`. Place chunk options within curly braces, after `"{r}"` inside with `"{r, echo=TRUE}"`  
`getRversion()`  
## [1] "3.2.3"

**GLOBAL OPTIONS**  
Set with `knitr::opts_chunk$set()`, e.g.  
`"{r, include=FALSE}"`  
`knitr::opts_chunk$set(echo=TRUE)`

### IMPORTANT CHUNK OPTIONS

**cache** - cache results for future knits (default = FALSE)  
**cache.path** - directory to save cached results in (default = ".cache/")  
**child** - file(s) to knit and then include (default = NULL)  
**collapse** - collapse all output into single block (default = FALSE)  
**comment** - prefix for each line of results (default = "##")

**dependson** - chunk dependencies for caching (default = NULL)  
**echo** - Display code in output document: (default = TRUE)  
**engine** - code language used in chunk (default = "R")  
**error** - Display error messages in doc (TRUE) or stop rendering when errors occur (FALSE) (default = FALSE)  
**eval** - Run code in chunk (default = TRUE)

**fig.align** - 'left', 'right', or 'center' (default = 'left')  
**fig.cap** - figure caption as character string (default = NULL)  
**fig.height, fig.width** - Dimensions of plots in inches  
**highlight** - highlight source code (default = TRUE)  
**include** - Include chunk in doc after rendering (default = TRUE)

**message** - display code messages in document (default = TRUE)  
**results** (default = "markup") as 's' - passthrough results  
**hide** - do not display results  
**tidy** - tidy code for display (default = FALSE)  
**warning** - display code warnings in document (default = TRUE)

Options not listed above: R.options, ani.opts, ani.width, background, cache.comments, cache.lazy, cache.results, cache.vars, dev, dev.args, epi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lc, fig.pos, fig.ppt, fig.ppt.opts, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, print, rst.label, render.size, split, tidy.opts



RStudio® is a trademark of RStudio, Inc. • CC BY-SA RStudio • info@rstudio.com • 344-448-1212 • rstudio.com • Learn more at [rmarkdown.rstudio.com](http://rmarkdown.rstudio.com) • rmarkdown 1.6 • Updated: 2015-02

## .rmd Structure



**YAML Header**  
Optional section of header (e.g., pandoc) option is written as key-value pairs (YAML).  
At start of file  
Between lines of  
**text**  
Narration formatted with markdown, mixed with:  
**Code Chunks**  
Chunks of embedded code. Each chunk:  
Begins with `"{r}"`  
ends with `"{r}"`  
R Markdown will run the code and append the results to the doc.  
It will use the location of the Rmd file as the **working directory**

## Parameters

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

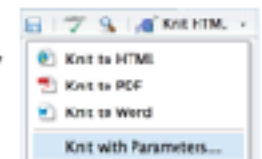
1. **Add parameters** - Create and set parameters in the header as sub-values of params

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

2. **Call parameters** - Call parameter values in code as `params$<name>`

```
Today's date:
r params$d
```

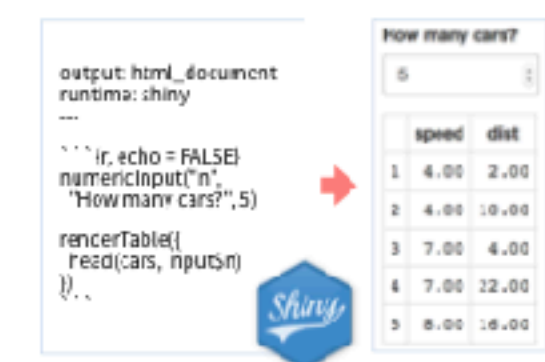
3. **Set parameters** - Set values with knit with parameters or the params argument of render().  
`render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01")))`



## Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

1. Add runtime - shiny to the YAML header
2. Call Shiny input functions to embed input objects.
3. Call Shiny render functions to embed reactive output.
4. Render with `render::render()` or click Run Document in RStudio IDE



Embed a complete app into your document with shiny: `shinyAppDir()`

NOTE: Your report will be rendered as a Shiny app, which means you must choose an HTML output format, like `html_document`, and serve it with an active R Session

# rstudio.com/cheatsheets