Introduction to Data Science Using R

Lecture 4 - Introduction to R Markdown

Ben Rogers, Sept 15, 2022

Review

Last Class

- Lists
- Dataframes
- Functions
- Scripts

Today

- Introduction to R Markdown!!

First, a review

R script

- Create a script in RStudio and do the following step by step

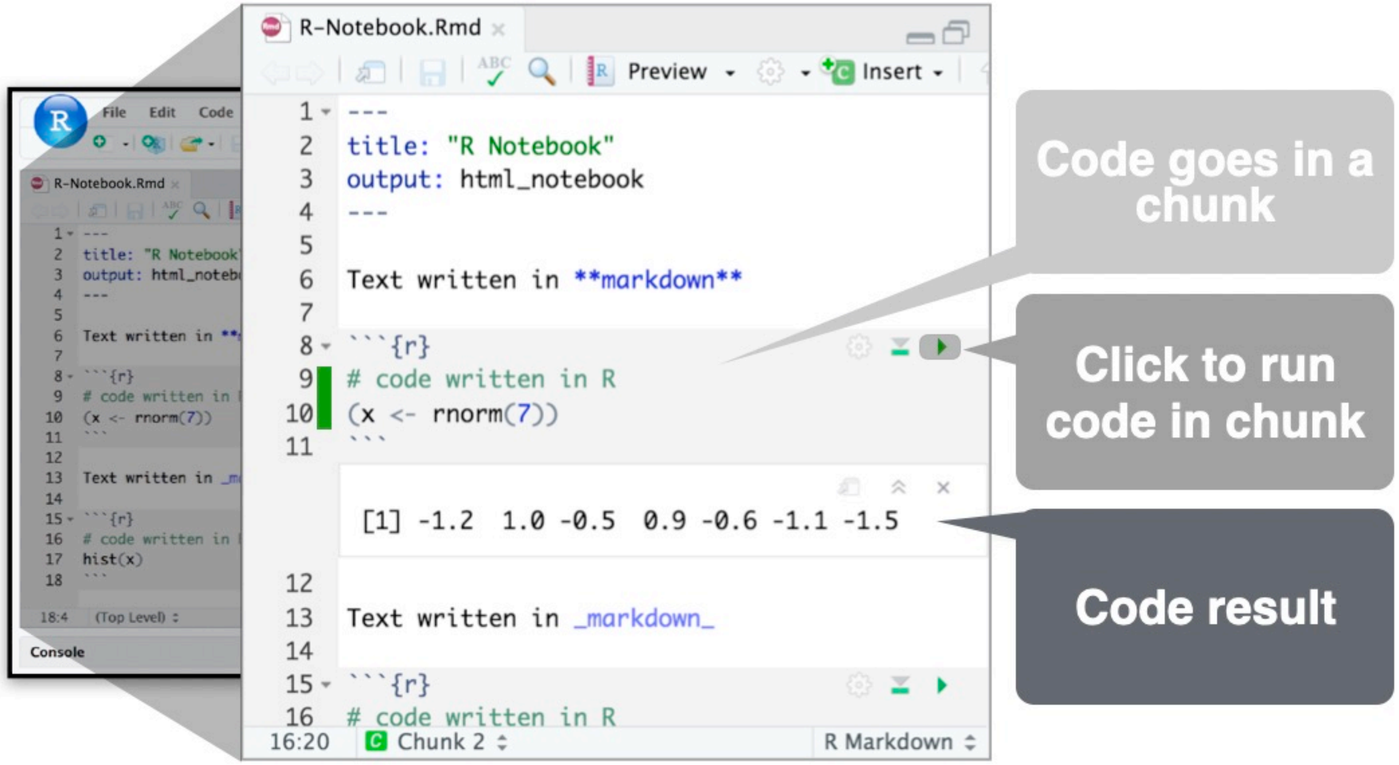
1. Calculate $\frac{\pi^2}{6}$ and save it in x.
2. Calculate $\frac{1}{1} + \frac{1}{2^2} + \cdots + \frac{1}{10^2}$
3. Take the difference between the two numbers obtained from the last two steps, save it in d1
4. Replace 10 with 20 in step 2 and do step 3 again, save the difference in d2
5. Replace 20 with larger numbers, do the same calculation and save them in d3, d4, d5, …
6. Combine d1, d2, d3, … in a vector d, did you find anything interesting in d?

Errors, warnings and messages

- R reports errors, warnings and messages in a glaring red font, which makes it seem like it is scolding you. R will show red text in the console pane in three different situations.
  - **Errors**: Generally when there's an error, the code will not run
  - **Warnings**: Generally, when there's a warning, your code will still work, but there may be something wrong.
  - **Messages**: When the red text doesn't start with either "Error" or "Warning", it's just a friendly message

Introduction to Markdown

R Markdown


R-Notebook.Rmd
Preview
Insert
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 ```
[1] -1.2 1.0 -0.5 0.9 -0.6 -1.1 -1.5
12
13 Text written in _markdown_
14
15 ```{r}
16 # code written in R
16:20 Chunk 2 R Markdown
Code goes in a chunk
Click to run code in chunk
Code result

R Markdown

- R Markdown files are designed to be used in three ways:

    1. For communicating to decision makers, who want to focus on the conclusions, not the code behind the analysis.

    2. For collaborating with other data scientists, who are interested in both your conclusions, and how you reached them (i.e. the code).

    3. As and environment in which to do data science, as a modern day lab notebook, where you can capture not only what you did, but also what you were thinking.

R Markdown tools and references

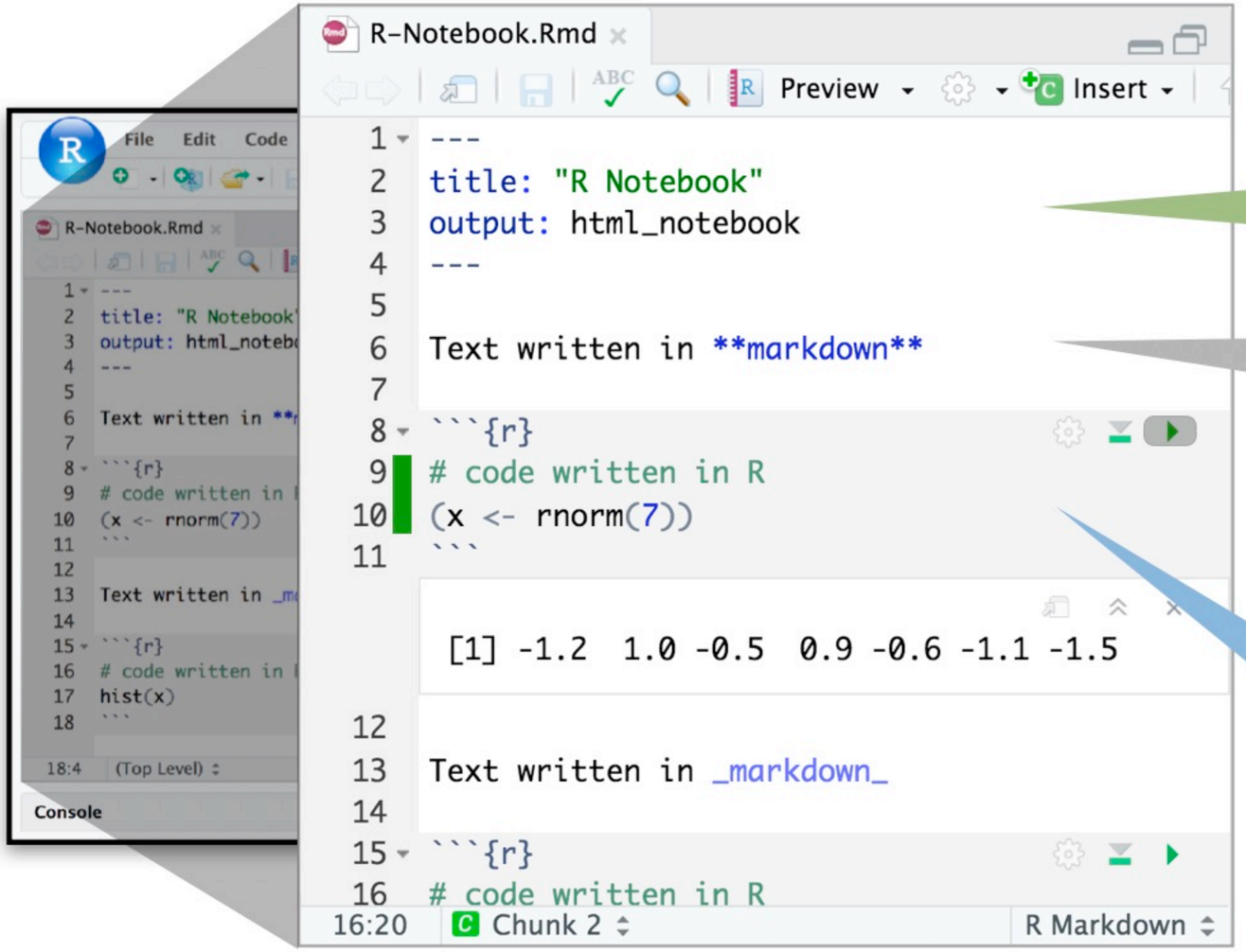
- R Markdown integrates a number of R packages and external tools. When you use R Markdown, keep these resources close at hand:
  - R Markdown Cheat Sheet: Help -> Cheatsheets -> R Markdown Cheat Sheet
  - R Markdown Reference Guide: Help -> Cheatsheets -> R Markdown Reference Guide

Getting started in R Markdown

- To use R Markdown, you will need to install the rmarkdown package:
    - install.packages(“rmarkdown”)
- In RStudio, go to File -> New File -> R Markdown
- Click “Knit” button

Create an R Markdown File

Plain text file with 3 types of content:



A YAML header surrounded by ---

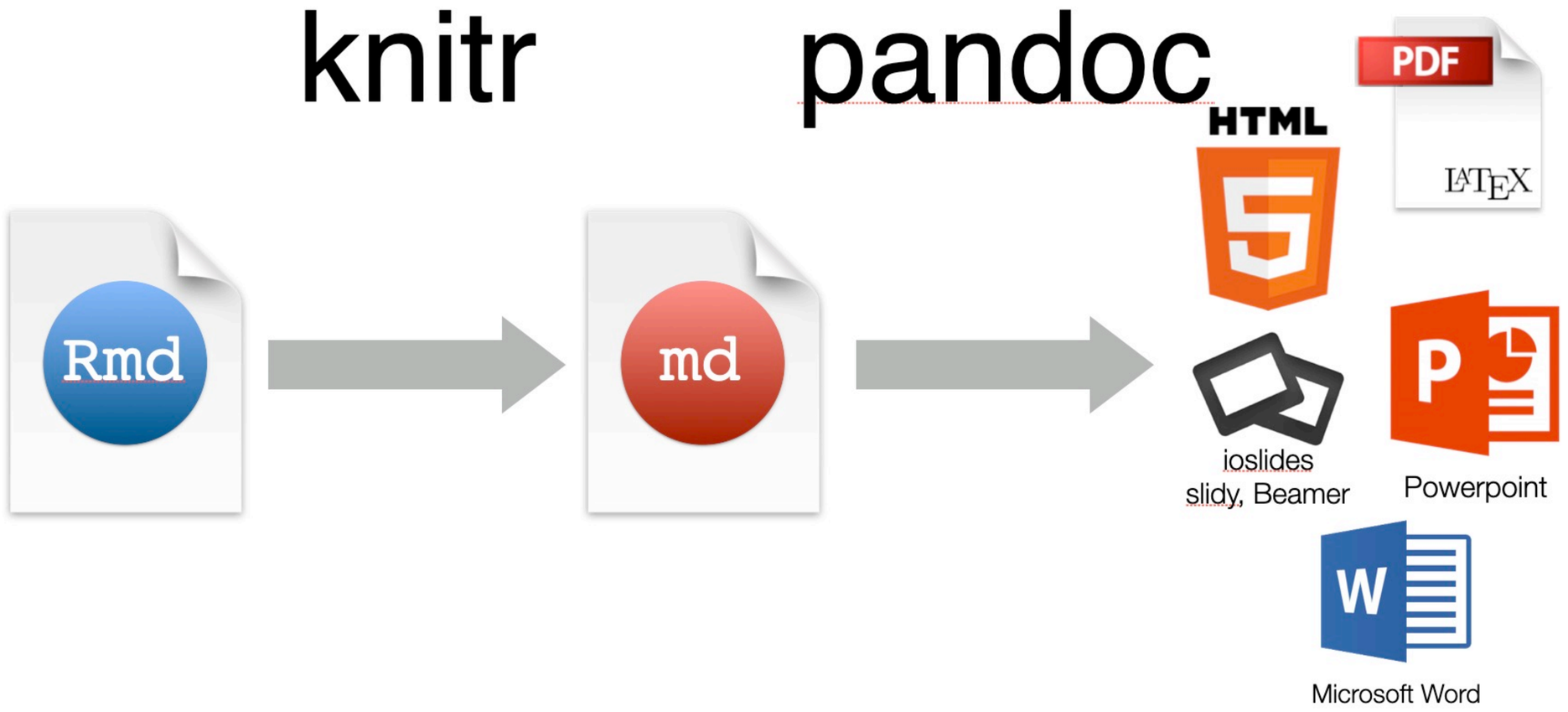
Text in markdown

Code chunks surrounded by ```

R Markdown

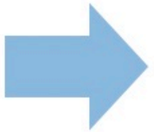
- You can run each code chunk by clicking the Run icon, or by pressing Cmd/Ctrl + Shift + Enter
    - RStudio executes the code and displays the results in line with the code.
- To produce a complete report containing all text, code, and results, click “Knit” or press Cmd/Ctrl + Shift + K.
    - This will display the report in the viewer pane, and create a self-contained HTML file that you can share with others.

Knitting an R Markdown file



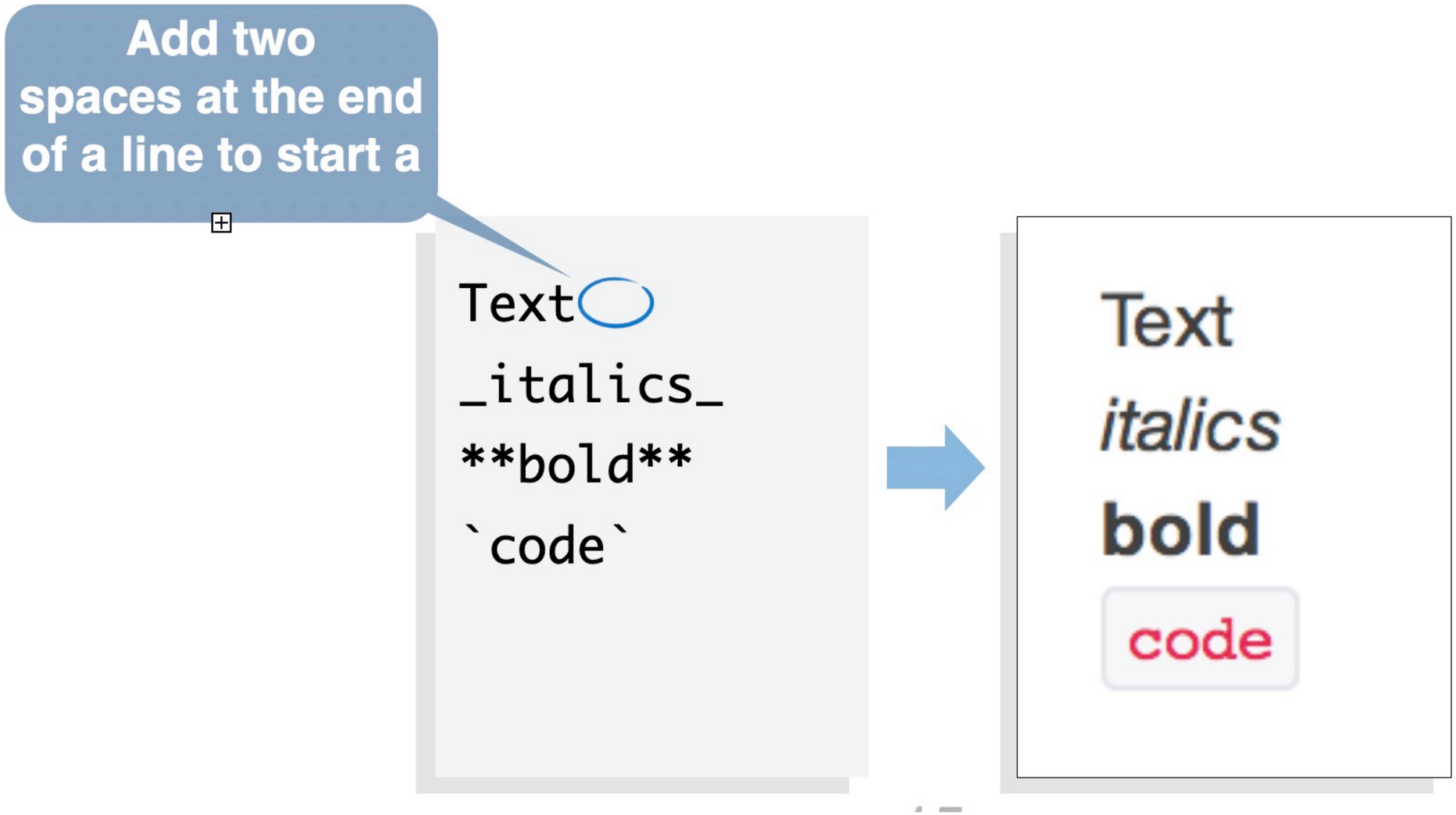
Text formatting with Markdown

```
# 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
Text
italics
bold
`code`
Text
italics
bold
code

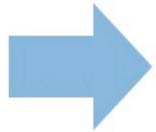
Lists

```
Bullets

* bullet 1
* bullet 2

Numbered list

1. item 1
2. item 2
```



Bullets

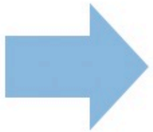
- bullet 1
- bullet 2

Numbered list

1. item 1
2. item 2

Links

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



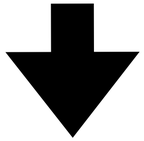
This is a link.

Images

```
![](images/1.png)
The RStudio logo.
```

Tables

```
First Header  | Second Header
------------- | -------------
Content Cell  | Content Cell
Content Cell  | Content Cell
```



First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

Your turn

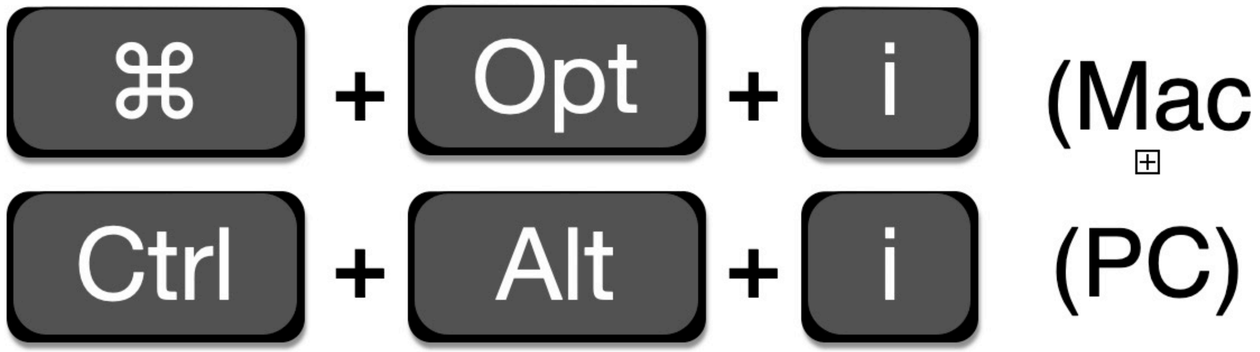
Using the R Markdown Quick Reference…

- Add a block quote.
- Add a horizontal rule.
- Add a subscript.

Code Chunks

- To run code inside an R Markdown document, you need to insert a chunk. There are four ways to do so:
    - The keyboard shortcut is Cmd/Ctrl + Alt + I
    - The “Insert” button icon in the editor toolbar
    - Code -> Insert Chunk in the menu bar
    - By manually typing the chunk delimiters

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



Chunk name

- Chunks can be given an optional name: ```` ```{r by-name} ````
- This has three advantages:
    - You can more easily navigate to specific chunks using the drop-down code navigator in the bottom-left of the script editor.
    - Graphics produced by the chunks will have useful names that make them easier to use elsewhere.
    - You can set up networks of cached chunks to avoid re-performing expensive computations on every run.

Chunk Options

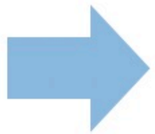
- By default, R Markdown includes both the code and its results.

Chunk Options

Chunk output can be customized with options — arguments supplied to chunk header

echo = FALSE hides the code.

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



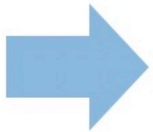
Here's some code

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

eval=FALSE prevents code from being run. No results will be displayed.

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

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



Here's some code

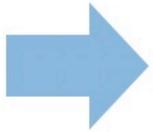
```
dim(iris)
```

include=FALSE runs the code, but prevents the code and the results from appearing.

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

````
Here's some code

```{r eval=FALSE}
dim(iris)
```
````

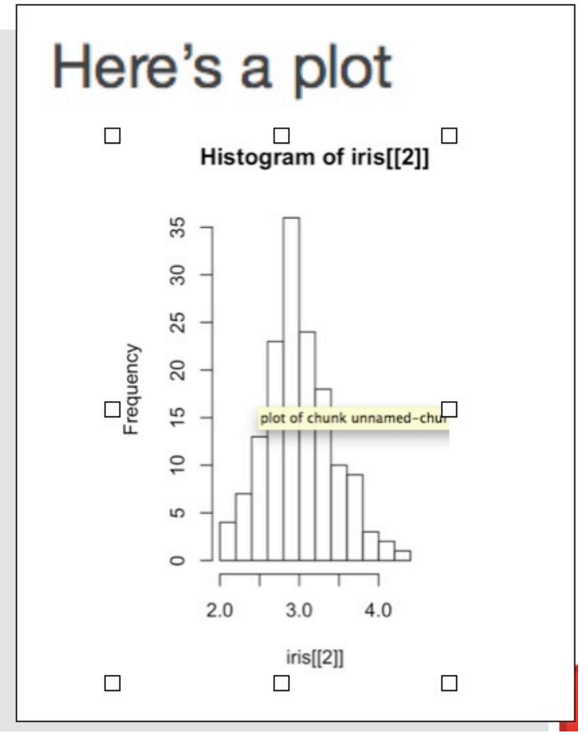


Here's some code

```
dim(iris)
```

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

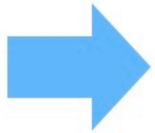
````
Here's a plot
```{r echo=FALSE, fig.width=3, fig.height=5}
hist(iris[[2]])
```
````



Option	Run code	Show code	Output	Plots	Messages	Warnings
`eval = FALSE`	-		-	-	-	-
`include = FALSE`		-	-	-	-	-
`echo = FALSE`		-				
`results = "hide"`			-			
`fig.show = "hide"`				-		
`message = FALSE`					-	
`warning = FALSE`						-

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.

Caching

- Normally, each knit of a document starts from a completely clean slate.
- This can be painful and slow if you have some computations that take a long time.
- You can use `cache = TRUE`
- When set, this will save the output of the code chunk to a specially named file on disk.
    - On subsequent runs, knits will check to see if the code has changed, and it it hasn't, it will reuse the cached results.

Your Turn

In your R Markdown script, add the following two code chunks:

````
```{r a}
x <- 1
```


```{r b,cache = TRUE}
y <- x + 2
y
```
````

1) Run the code, what is the value of y?
2) Change to , what is the value of y?
3) Change to , what is the value of y?

Explain the result.

Global options

- You can change the default options for code chunks by calling `knit::opts_chunk$set()` in a code chunk.

- The following code with hide code by default, so you only show the code that you deliberately choose to (with echo = TRUE)

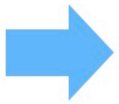
```
Knitr::opts_chunk$set(echo=FALSE)
```

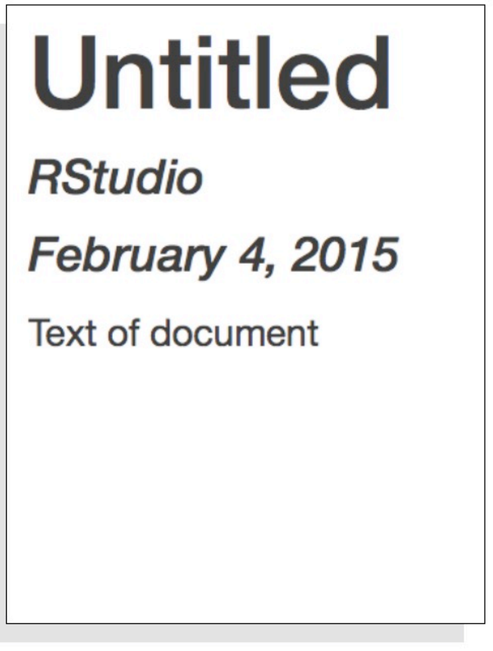
YAML header

- You can control many other “whole document” settings by tweaking parameters in the YAML header.
- YAML : “Yet another markup language”

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

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





Numbered headers

- You can create an R markdown file with numbered sections using the following YAML header. Be careful about the indent.

```
---
title: "Problem set 1"
author: ""
date: ""
output:
  html_document:
    number_sections: true
---
```

Have a good weekend!