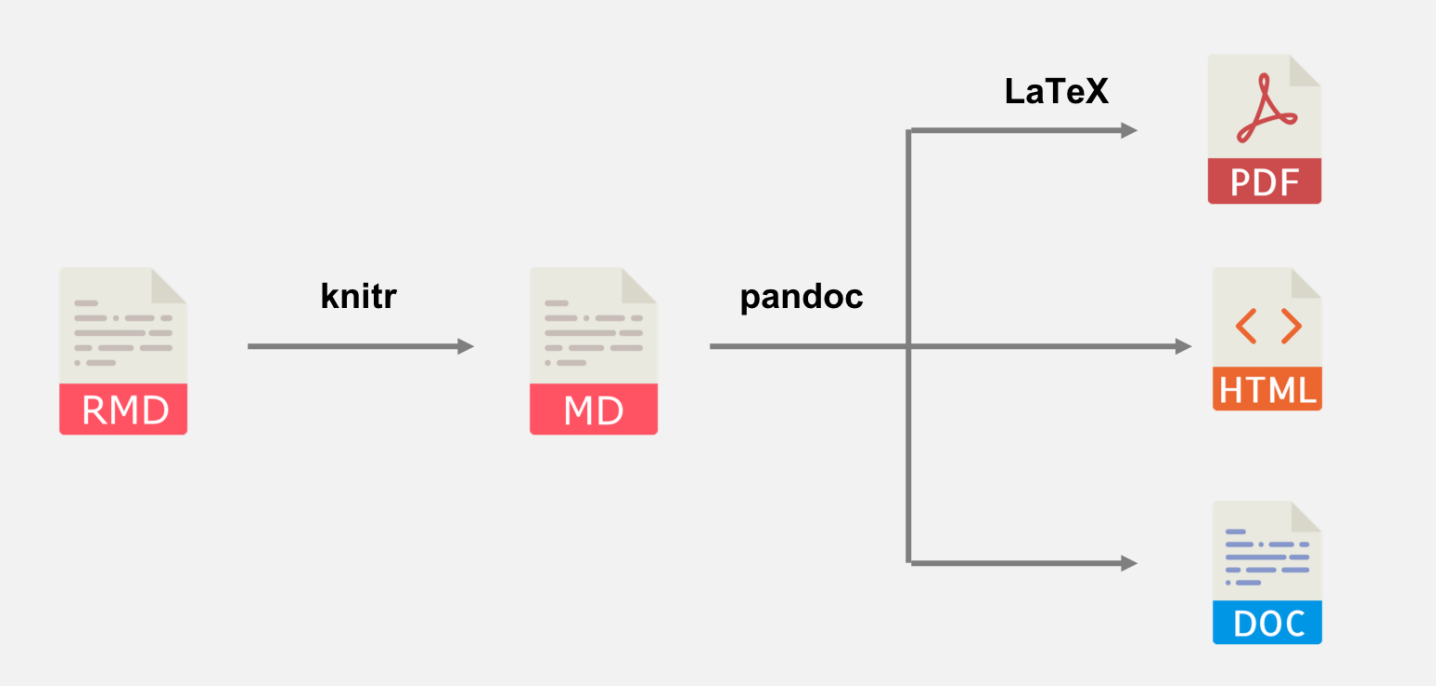
Description of R Markdown

- Uses Markdown syntax
- .Rmd file extension instead of .R
- Supports multiple output formats (.html, .pdf, Word documents, and others)
- Can display R code along with R output

R Markdown process



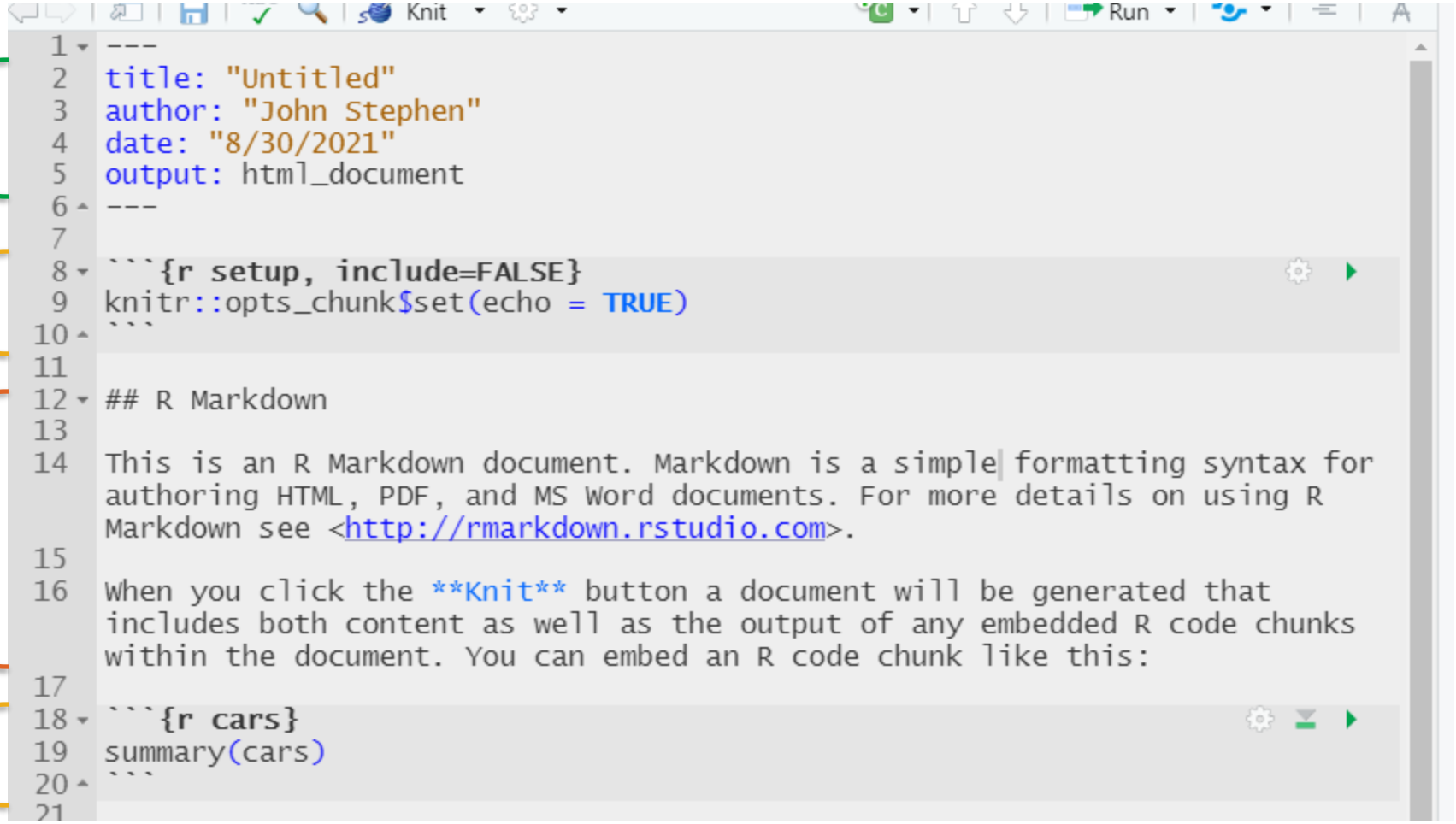
R Markdown Cookbook
Yihui Xie, Christophe Dervieux, Emily Riederer
2021-08-17

Sections of R Markdown file

Header
YAML metadata

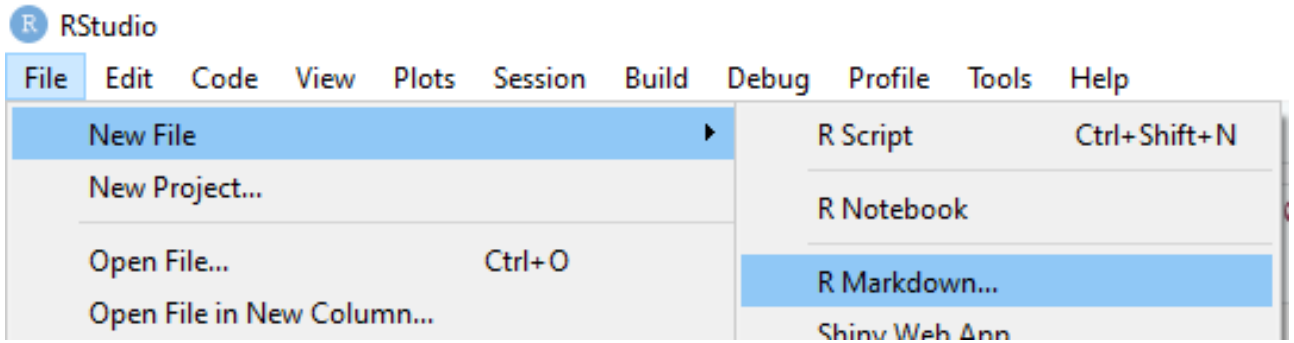
“Chunk”
R Code that will be ran

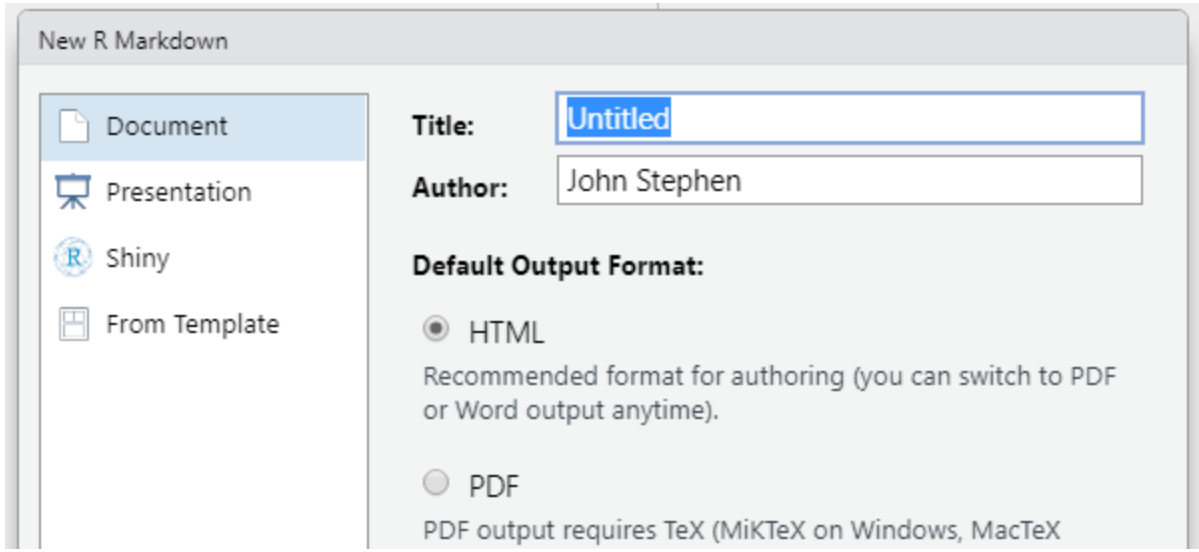
Markdown text



Creating a new R Markdown file

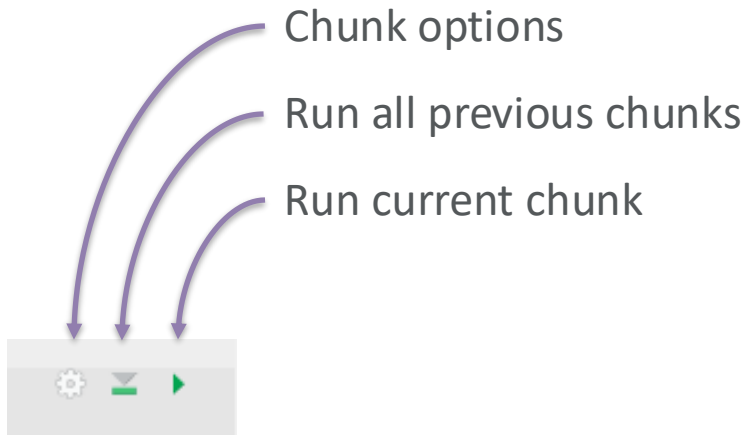
- Can create a new R Markdown file using the file menu
- The new R Markdown window lets you choose title of document, format, etc.
  - These options get entered into the YAML header





Running R Markdown interactively

- Can run the sections of code (“chunks”) interactively before creating your final output
- Output is then displayed below chunk
- Note: Sometimes your final output won’t look like output created interactively



“Knitting” R Markdown document

- Creating R Markdown output is called “knitting” the document
- Output is created by
  - Clicking on the knit icon on the toolbar
  - Or running the following code:
    - **rmarkdown::render()**
  - Example of knitting an R Markdown document using code:
    - **rmarkdown::render(input = 'Exercise_4.Rmd', output_format = 'html_document')**

Output formats

- R Markdown natively supports HTML, PDF, and Microsoft Word formats
- You can control your output format:
  - Through the wizard when you first create an R Markdown file
  - In the YAML header
- Note: You need LaTeX install to create pdf output.
- Can install with **tinytex::install_tinytex()**

```
---
title: "Untitled"
author: "John Stephen"
date: "9/19/2021"
output: word_document
---
```

```
---
title: "Untitled"
author: "John Stephen"
date: "9/19/2021"
output: html_document
---
```

```
---
title: "Untitled"
author: "John Stephen"
date: "9/19/2021"
output: pdf_document
---
```

Code “Chunks”

- R chunks are sections that contain R code
- Start with ```**{r}** and ends with ```
  - (Note these are back ticks, not single quotes)

Code:

````
35
36 ```{r cars}
37
38 head(mtcars) %>%
39   kable()
40
41 ```
42
````

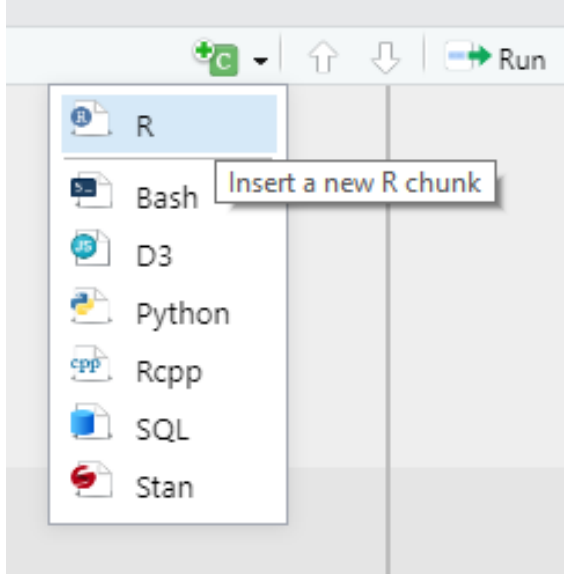
HTML output:

```
head(mtcars) %>%
  kable()
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

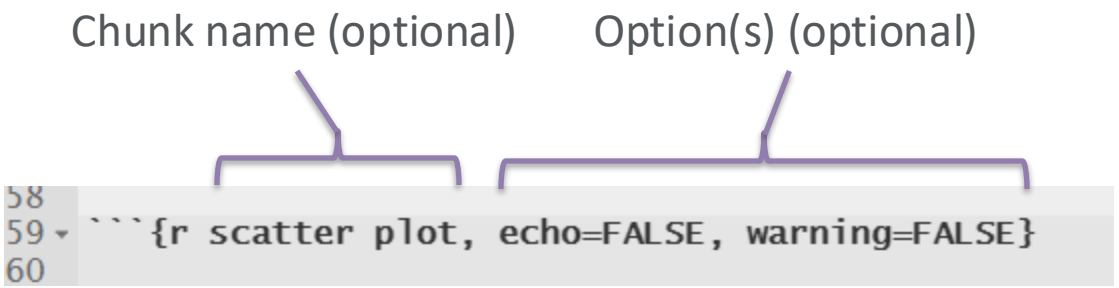
Code “Chunks” (cont.)

- Multiple ways to insert an r chunk
  - Use icon on toolbar
  - Type manually with starting with ```**{r}** and ending with ```
  - Windows hotkey: Ctrl + Alt + i
  - Mac hotkey: Ctrl + Option + i



Code “Chunks” (cont.)

- Chunks can contain a name, and several options
- If chunk names are used, they need to be unique
- Use commas to separate options (if used)



“Chunk” options

- Examples of chunk options:

Option:	Default:	Effect:
eval	TRUE	Runs code in chunk
include	TRUE	Shows or hides R code and results in output
echo	TRUE	Shows or hides R code in output
error	FALSE	Controls whether R stops on error
warning	TRUE	Displays warning
message	TRUE	Displays message

Setting default options for chunks

- Use the **opts_chunk$set** function to set default chunk options for your document
- You can still change these options later in the document

```
9  knitr::opts_chunk$set(echo = TRUE)
10
```

Will display R code in output.

```
12 knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
13
```

Will display R code, suppress warnings, and suppress messages in output.

Inline R code

- Inline code lets you evaluate R code without a chunk.
- The results will then be displayed with the text.
- Inline code will display the results, but not the code.
- Cannot use chunk options.
- Starts with `` `r ``
- Ends with `` ` ``

Code:

```
You can use this to enter things like sample sizes (n: `r nrow(mtcars)`) into
your text, the current date `r Sys.Date()` into your text, etc...
```

Result:

You can use this to enter things like sample sizes (n: 32) into your text, the current date 2021-09-17 into your text, etc…

Markdown syntax

Code:

```
81 # This is a header
82
83 You can create non numbered lists:
84
85 - Item 1
86 - Item 2
87
88 And numbered lists:
89
90 1. Item 1
91 2. Item 2
92
93 ## This is a sub header
94
95 You can also **Bold** or *italicize* text.
96
```

Output:

This is a header

You can create non numbered lists:

- Item 1
- Item 2

And numbered lists:

1. Item 1
2. Item 2

This is a sub header

You can also **Bold** or *italicize* text.

Markdown syntax (cont.)

- **Code:**

```
## More syntax examples

Footnotes[^1] are also supported.

[^1]: Footnote text will appear at end of document.

You can also enter a link to a web-site:
[Northwestern](http://www.northwestern.edu "Northwestern").
```

- **Output:**

More syntax examples

Footnotes[1] are also supported.

You can also enter a link to a web-site: Northwestern.

1. Footnote text will appear at end of document.↩

Table of contents

- Create a table of contents by specifying it in the YAML header
- R Markdown will populate your table of contents with your headers that are in your document
  - (Text that starts with #'s)

YAML header for HTML output:

```
---
title: "Presentation"
author: "John Stephen"
date: "8/30/2021"
output:
  html_document:
    toc: yes
---
```

YAML header for HTML output:

```
---
title: "Presentation"
author: "John Stephen"
date: "8/30/2021"
output:
  html_document:
    toc: yes
    toc_float: yes
---
```

YAML header for PDF output:

```
---
title: "Presentation"
author: "John Stephen"
date: "8/30/2021"
output:
  pdf_document:
    toc: yes
---
```

Table of contents (cont.)

- R Markdown will take headers in your document and create a table of contents
- Table of contents will nest headings based on the number of hash marks (#)

PDF output

Creating tabs (html output)

- Can group output in tabs
- Start tab section by entering **{.tabset}** in parent header
- Subsequent sub headers will appear as a tab
- Output that is under each sub header will go into corresponding tab
- End tabs by entering **{ - }** in a header

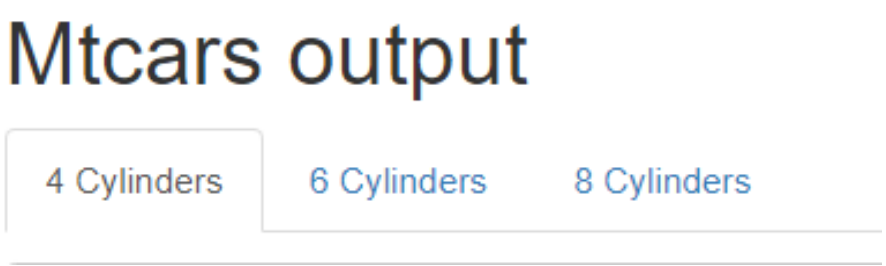
Start:

```
20 # Mtcars output {.tabset}
21
22 ## 4 Cylinders
```

End:

```
58
59 # { - }
60
```

HTML output:



Additional resources

- **R Markdown Cookbook**
  - https://bookdown.org/yihui/rmarkdown-cookbook/
- **R Markdown: The Definitive Guide**
  - https://bookdown.org/yihui/rmarkdown/
- **R Studio Cheatsheets**
  - https://www.rstudio.com/resources/cheatsheets/
- **R Studio R Markdown tutorial**
  - https://rmarkdown.rstudio.com/lesson-1.html

Exercise #1

- First we'll try creating output with text and some output of R code. Use the dataset **mtcars** for this exercise
- **mtcars** is included with R, so you don't have to download the dataset
  - (Can load into environment using **mtcars <- get(data(mtcars))**)
- **Try the following:**
  - Create an R Markdown file.
  - Write R code to look at the first few rows of the **mtcars** dataset.
    - (use **head()** function)
  - Try writing some text to accompany your output. Try formatting your markdown text where appropriate.
  - Knit your markdown file to html.

Exercise #2

- Now we'll try creating a plot and suppressing the code.

- **Try the following:**
  - Create a few plots looking at any associations that look interesting.
  - Try suppressing the code for your plots, so that your report just has plots.
  - Also try using inline code to integrate r results into your text.
    - (you can look at numbers of rows and columns using **nrow()** and **ncol()**).
  - Knit to html.

Exercise #3

- Next we'll try creating a pdf document
- We'll also add a table of contents
- Note: if Latex is not installed on your computer, you may have to run **tinytex::install_tinytex()**
- **Try the following:**
  - Create a few plots looking at any associations that look interesting.
  - Make sure to create headers for each plot. (Using #'s before your text).
  - Try suppressing the code for your plots, so that your report just has plots.
    - Instead of setting the options in each chunk, set the default to suppress code.
  - Try adding a table of contents.
  - Knit to pdf.

Exercise #4

- Finally we'll creating a document with tabs
- **Try the following:**
  - Think of a categorization you may want to use for some of the output you have created so far.
  - Try creating tabs to categorize the output you have created so far (or create new output). Put them into the appropriate tab.
  - Knit to html.