Creating Reproducible Analyses: Introduction to RMarkdown

Grace J. Goodwin
(she/ her/ hers)
@gracejgoodwin

Slides adapted from David Keyes (@dgkeyes), inspired by Danielle Navarro (@djnavarro) and Paul Campbell (@paulcampbell91)

Art by @allison_horst

What is RMarkdown?

What is RMarkdown?

Authoring framework for data science.

What is RMarkdown?

Authoring framework for data science.

You can:

1. Write, save, and run code
2. Generate high-quality reports that can be shared with an audience

RMarkdown Interface

What can I use RMarkdown for?

What can I use RMarkdown for?

- Share your analyses and results with your lab in a variety of formats (e.g., PDF, HTML, Word)
- Build interactive applications (e.g., Shiny)
- Write journal articles
- Make slides for presentations (like this one!)
- Create websites or blogs
- and more!

Why use RMarkdown?

Why use RMarkdown?

- RMarkdown was designed for easier **reproducibility**.

Why use RMarkdown?

- RMarkdown was designed for easier **reproducibility**.
- Your code and narrative are in the same place, so collaborators can see what you did, why you did it, and how you obtained your results.

Why use RMarkdown?

- RMarkdown was designed for easier **reproducibility**.
- Your code and narrative are in the same place, so collaborators can see what you did, why you did it, and how you obtained your results.
- RMarkdown is flexible and can support several languages (e.g., Python, C++, Java, etc.) and several output types, which makes collaboration easy.

Why use RMarkdown?

- RMarkdown was designed for easier **reproducibility**.
- Your code and narrative are in the same place, so collaborators can see what you did, why you did it, and how you obtained your results.
- RMarkdown is flexible and can support several languages (e.g., Python, C++, Java, etc.) and several output types, which makes collaboration easy.

You can alternate between text and code within the same document

```
#you don't need to use hashmarks for text (outside of a chunk)
```

You can also visualize what your code will look like once it's knitted!

RMarkdown Overview

RMarkdown Overview

Every RMarkdown document has the following:

YAML

```yaml
---
title: "This workshop is awesome"
author: "Joscelin Rocha Hidalgo"
date: "07/18/2020"
output: word_document
---
```

YAML

```yaml
---
title: "This workshop is awesome"
author: "Joscelin Rocha Hidalgo"
date: "07/18/2020"
output: word_document
---
```

Stands for "YAML Ain't Markup Language"

YAML

```
---
title: "This workshop is awesome"
author: "Joscelin Rocha Hidalgo"
date: "07/18/2020"
output: word_document
---
```

Stands for "YAML Ain't Markup Language"

The very top of your RMarkdown

YAML

```
---
title: "This workshop is awesome"
author: "Joscelin Rocha Hidalgo"
date: "07/18/2020"
output: word_document
---
```

Stands for "YAML Ain't Markup Language"

The very top of your RMarkdown

Where you add title, author, date, output options, etc.

YAML

```
---
title: "This workshop is awesome"
author: "Joscelin Rocha Hidalgo"
date: "07/18/2020"
output: word_document
---
```

Stands for "YAML Ain't Markup Language"

The very top of your RMarkdown

Where you add title, author, date, output options, etc.

Specify how you want your knitted file to look (e.g., do you want a table of contents? do you want your code to be visible? etc.)

Text


---
title: "My Super Fancy Report"
author: "David Keyes"
output: html_document

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
R Markdown
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r cars}
summary(cars)
```
Including Plots
You can also embed plots, for example:
```{r pressure, echo=FALSE}
plot(pressure)
```
Text
Text

Text

Markdown

```
Text with **some words in bold**
and *some words in italics*
```

Text

Markdown

```
Text with **some words in bold**
and *some words in italics*
```

Output

Text with **some words in bold** and *some words in italics*

Headers

Markdown

```
# First-Level Header

## Second-Level Header

### Third-Level Subheader
```

Headers

Markdown

```
# First-Level Header

## Second-Level Header

### Third-Level Subheader
```

Output

First-Level Header

Second-Level Header

Third-Level Subheader

Lists

Markdown

```
- Bulleted list item
- Bulleted list item
```

```
1. Numbered list item
1. Numbered list item
```

Lists

Markdown

```
- Bulleted list item
- Bulleted list item

1. Numbered list item
1. Numbered list item
```

Output

- Bulleted list item #1
- Bulleted list item #2

1. Numbered list item #1
2. Numbered list item #2

Inline Code

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

Inline Code

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

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

Inline Code

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

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

becomes

Inline Code

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

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

becomes

Two plus two equals 4.

Inline Code

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

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

becomes

Two plus two equals 4.

- This is great for writing papers!

Code Chunk

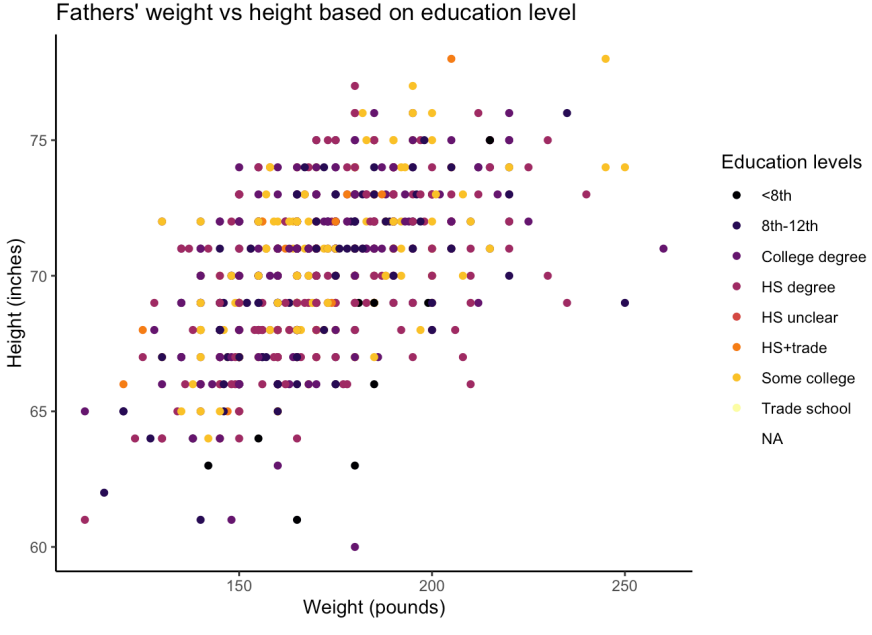
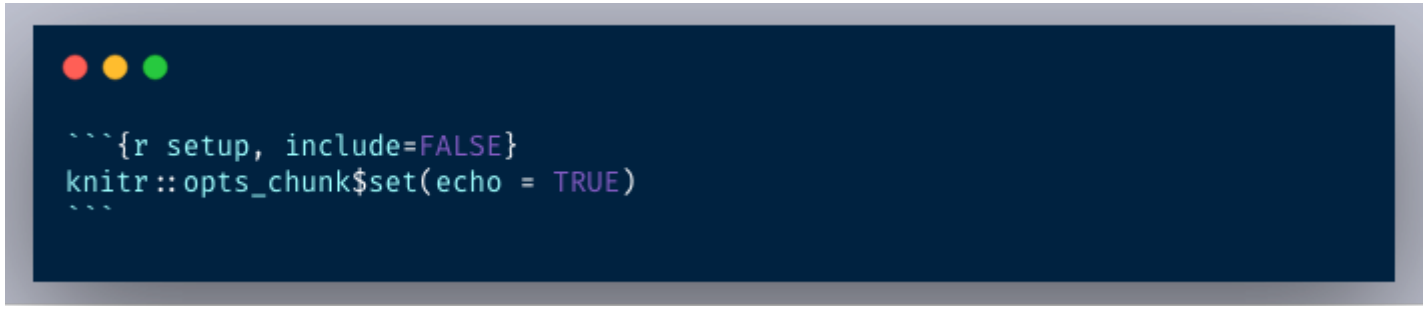
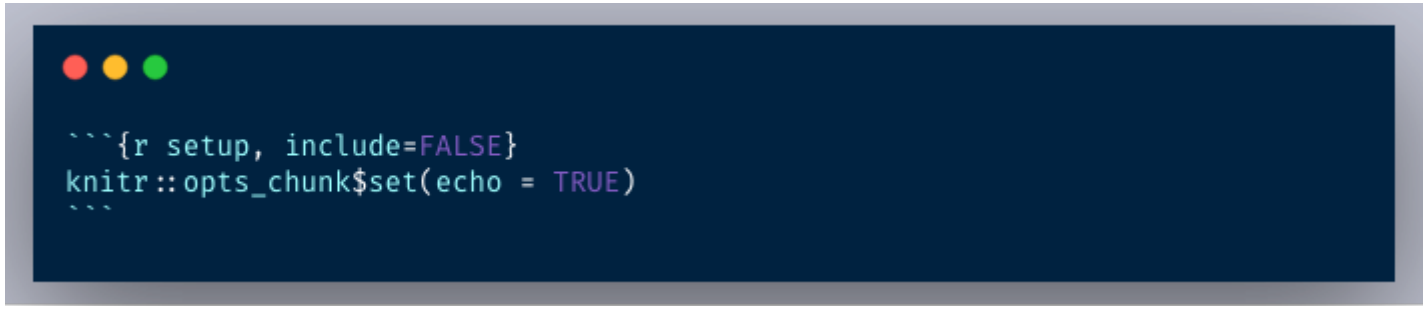
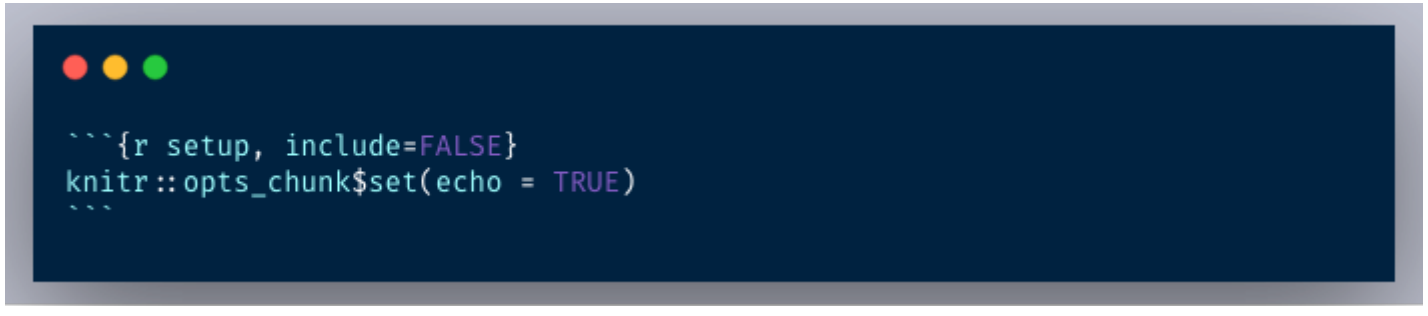
They start with three backticks and {r} and end with three backticks.

````
```{r}
ggplot(data,aes(dwt,dht, color = ded_lbls)) +
  geom_point() + scale_color_viridis_d(option = "inferno") +
  labs (title = "Fathers' weight vs height based on education level",
        x = "Weight (pounds)",
        y = "Height (inches)",
        color = "Education levels") +
  theme_classic()

```
````

Code Chunk

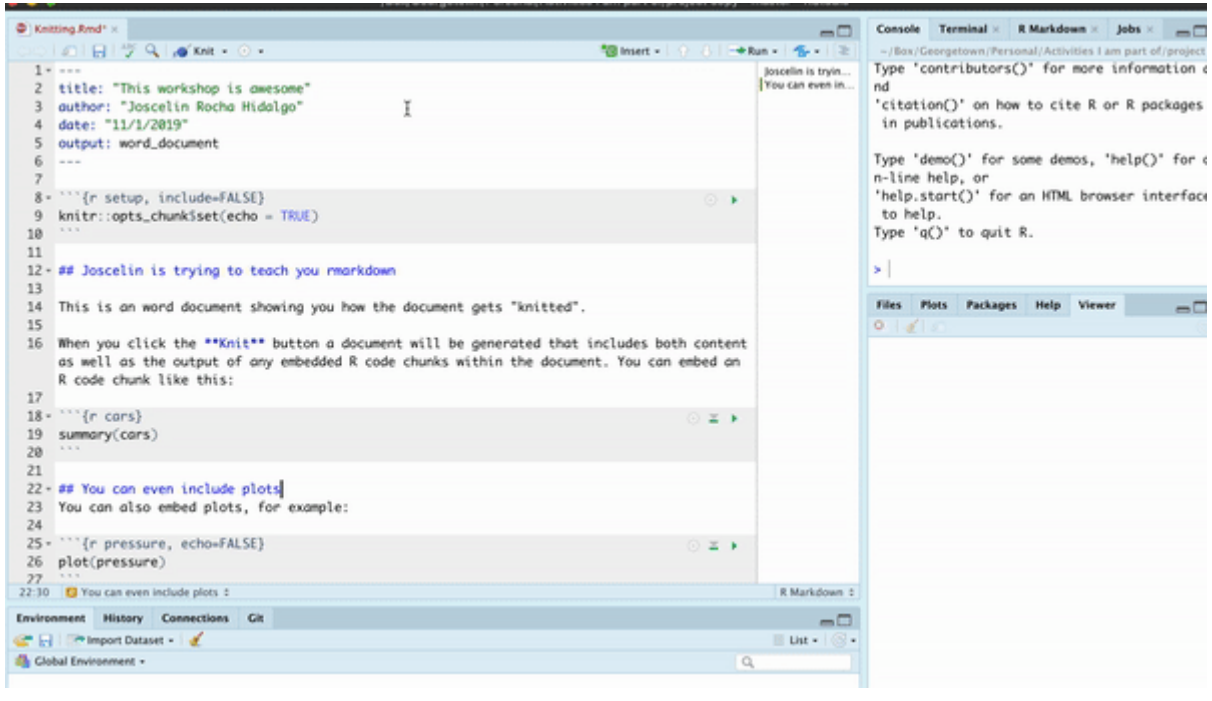
They start with three backticks and {r} and end with three backticks.

```
```{r}
ggplot(data,aes(dwt,dht, color = ded_lbls)) +
  geom_point() + scale_color_viridis_d(option = "inferno") +
  labs (title = "Fathers' weight vs height based on education level",
        x = "Weight (pounds)",
        y = "Height (inches)",
        color = "Education levels") +
  theme_classic()

```
```



Insert a Code Chunk: Button

Insert a Code Chunk: Keyboard Shortcut

Windows

control+alt+i

Insert a Code Chunk: Keyboard Shortcut

Windows

control+alt+i

Mac

command+option+i

Chunk Options

Other options that we won't discuss today:

- **`warning`** (show any warnings that R throws)
- **`message`** (show any messages that R sends)
- **`fig.width`** (default figure width)
- **`fig.height`** (default figure height)
- **`echo`** (show the R code in the knitted report)
- and many more ...

Setup Code Chunk

Setup Code Chunk

A special code chunk with the text `setup` right after the `r`.



Setup Code Chunk

A special code chunk with the text `setup` right after the `r`.



````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
````

All chunk options can be set at the **global level** (in the setup code chunk) or at the **chunk level** (for individual chunks).

Setup Code Chunk

A special code chunk with the text `setup` right after the `r`.



````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
````

All chunk options can be set at the **global level** (in the setup code chunk) or at the **chunk level** (for individual chunks).

Options at the individual chunk level **override** global chunk options.

Knitting (aka Export)



Your RMarkdown won't knit if you have errors in your code. Knit early and often to identify errors!

Writing Reproducible Code

(adapted from Harvard WiP Crash Course in R by Kirsten Morehouse)

Writing Reproducible Code

(adapted from Harvard WiP Crash Course in R by Kirsten Morehouse)

Reproducible = When you send someone your code and data, they can 100% reproduce your results, without getting errors (or divergent results)

- Make sure you are using the **most recent version of R** (sometimes packages are or aren't compatible with different versions of R)

- Make sure you are using the **most recent version of R** (sometimes packages are or aren't compatible with different versions of R)

- Update your packages: `update.packages()`

- Make sure you are using the **most recent version of R** (sometimes packages are or aren't compatible with different versions of R)
- Update your packages: `update.packages()`
- Load AND install packages:
  - `if (!require("packagename")) {install.packages("packagename"); require("packagename")}`

- Make sure you are using the **most recent version of R** (sometimes packages are or aren't compatible with different versions of R)
- Update your packages: `update.packages()`
- Load AND install packages:
  - `if (!require("packagename")) {install.packages("packagename"); require("packagename")}`
- DON'T include paths when you load data. Your path won't work for other people. Specify working directories instead.

- Make sure you are using the **most recent version of R** (sometimes packages are or aren't compatible with different versions of R)
- Update your packages: `update.packages()`
- Load AND install packages:
  - `if (!require("packagename")) {install.packages("packagename"); require("packagename")}`
- DON'T include paths when you load data. Your path won't work for other people. Specify working directories instead.
- Save any models you've run.

- Make sure you are using the **most recent version of R** (sometimes packages are or aren't compatible with different versions of R)
- Update your packages: `update.packages()`
- Load AND install packages:
  - `if (!require("packagename")) {install.packages("packagename"); require("packagename")}`
- DON'T include paths when you load data. Your path won't work for other people. Specify working directories instead.
- Save any models you've run.
- Clear your environment and try running everything again. Bonus if you run your code on a new machine.

Share your code

Info to include:

- README file
  - What should users expect to see in your repo?
- Instructions for reproducing your results (make sure to include the R script and data names, as well as the order in which the scripts shoudl be run)
- Data + Codebook
- Analysis script (bonus points for data cleaning script, too!)
- Saved models

Your Turn

1. Create a new RMarkdown file, setting the default output format as Word.
    - File > New File > R Markdown...
2. Save your RMarkdown file as report.Rmd.
3. Go into the YAML and change the title to “My 2022 Report.”
4. Change the output format to HTML by changing `output: word_document` to `output: html_document`.
5. Add the following first-level header: "Introduction"

03:00

Your Turn

6. Add this text (note the bold and italics) below the introduction header: "My name is (write your name here). I am the most **amazing** human being. You've *never* met someone like me. Please hire me!"
7. Add the following second-level header: "Reasons Why I am the Best"
8. Add the following list of reasons:

```
- Because I say so
- Because it is true
- Why would I lie?
```

9. Create a chunk using a shortcut
10. Calculate 2 + 2 and save the result as a variable called "my_var"
11. Print "my_var" using print()
12. Knit and reopen the report.html file

05:00

More Resources

More Resources (this just in!!)