



THE UNIVERSITY
of EDINBURGH |

Uusher
institute

HDS Tutorial 3

Week 6

| Brittany Blankinship | 17 & 19 May 2022 |



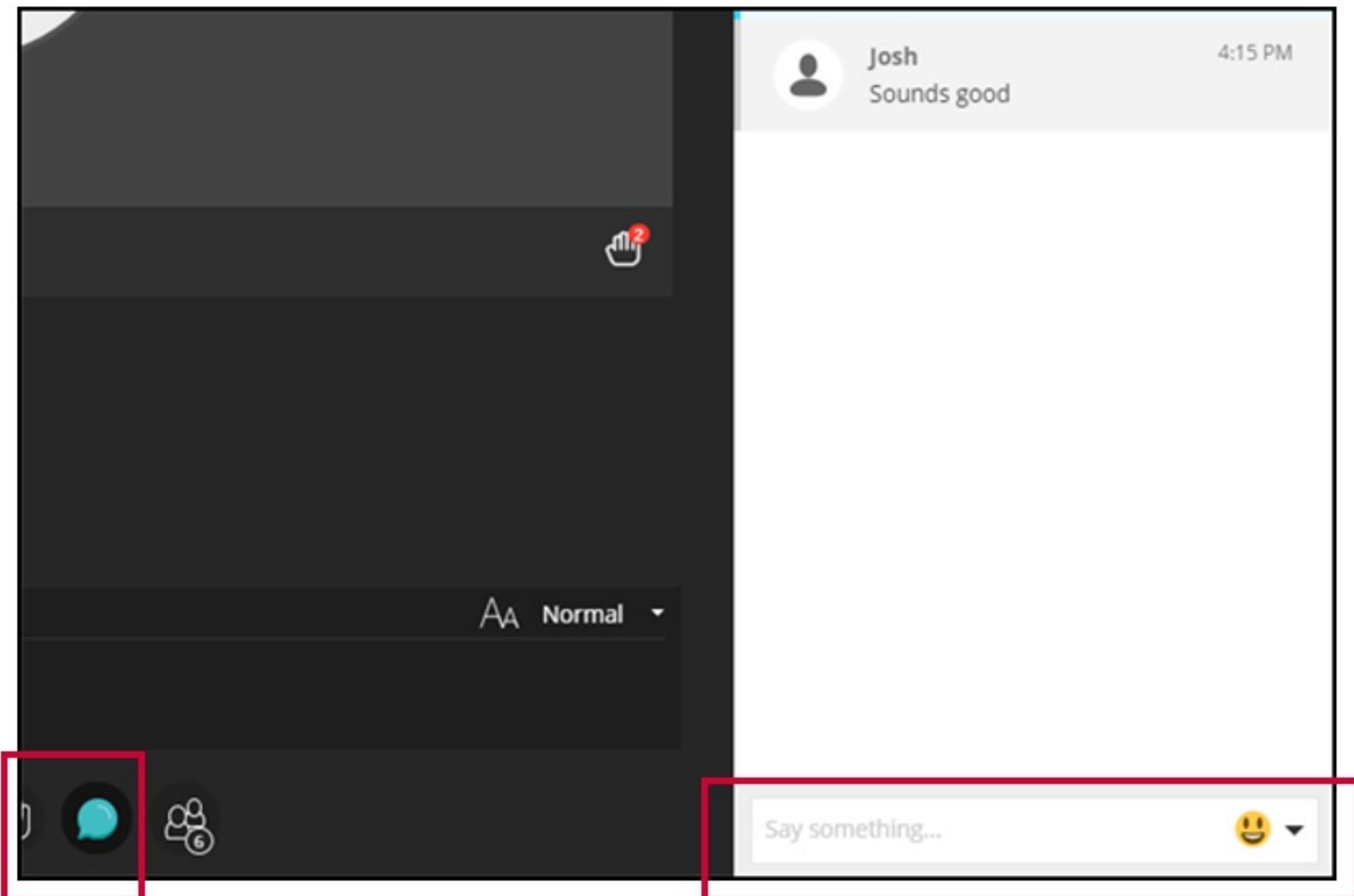
Audio check

Can you hear the presenter talking?

Please type **yes** or **no** in the “Text chat area”

If you can't hear:

- Check your Audio/Visual settings in the Collaborate Panel
- Try signing out and signing back into the session
- Type into the chat box and a moderator will try to assist you





THE UNIVERSITY
of EDINBURGH

| Uisher
Institute

Recording

Open to
the world

This session will now be recorded. Any further information that you provide during a session is optional and in doing so you give us consent to process this information.

These sessions will be stored by the University of Edinburgh for one year and published for 30 days after the event. Schools or Services may use the recordings for up to a year on relevant websites.

By taking part in a session, you give us your consent to process any information you provide during it.

Start Recording

ddi.hsc.talent@ed.ac.uk

Supported by



THE UNIVERSITY
of EDINBURGH

Data-Driven
Innovation



THE UNIVERSITY
of EDINBURGH |

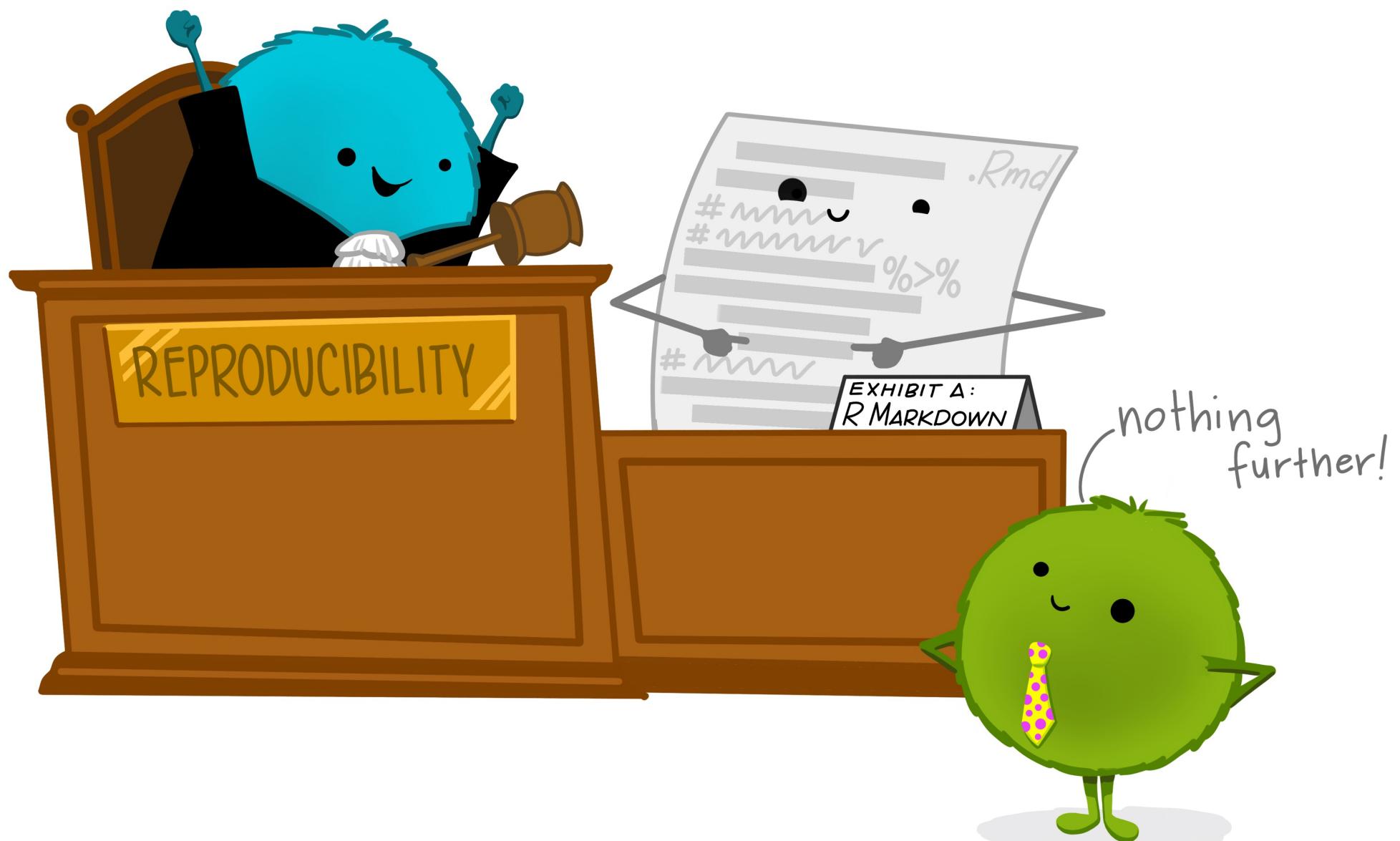
Uusher
institute

HDS Tutorial 3

Week 6

| Brittany Blankinship | 17 & 19 May 2022 |

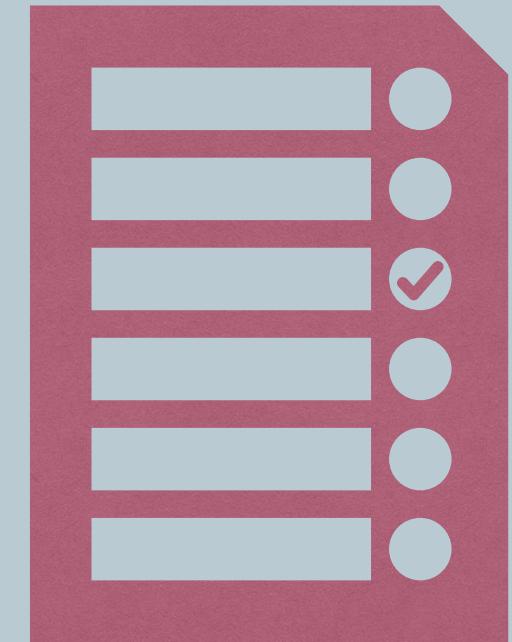
Agenda



Artwork by Allison Horst

@allison_horst

- **What is R Markdown**
- **Why use R Markdown**
- **Demonstration**
- **Resources for further study**
- **Q&A**



Have you installed all of the
necessary packages for
R Markdown?

Have you already looked through some
of this week's content
(i.e., opened the R Markdown practice
document?)

You might be thinking...

Okay...but why R Markdown,
why should we not just stick
with script files?

An R Markdown workflow is...

Less

- ↓ Error-prone
- ↓ Time consuming (once you get the hang of it)

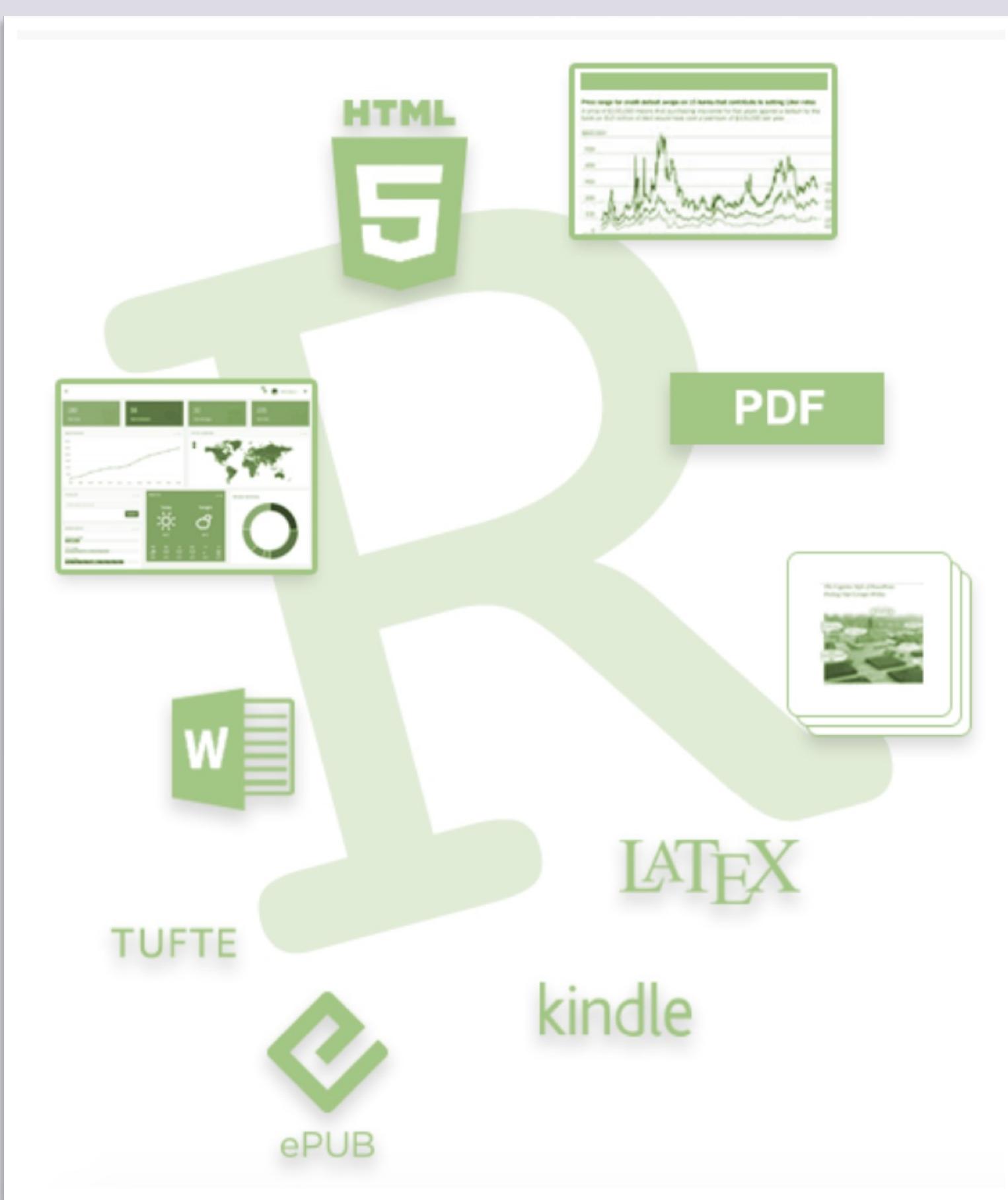
More

- ✓ Dynamic
- ✓ Reproducible
- ✓ Transparent



- Create dynamic documents that combine code, output (including figures + tables), and writing
- Same document holds code & narrative surrounding the data -- an authoring framework for data science!
- Can be used to...
 - Reproduce your analyses
 - Collaborate and share code with others
 - Communicate your findings with others (even those who do not understand code)

- Present analyses in high quality documents, reports, and presentations
- Support dozens of output formats, like PDFs, Word files, slideshows, and more
- Documents are fully reproducible — you can update your document at any time by re-knitting the code chunks
- Productive notebook interface weaves together narrative text & code to produce elegantly formatted output
- Can use multiple languages including R, Python, and SQL



R Markdown as...

1. Literate Programming
2. A Data Product
3. A Control Document
4. Templating

Credit: [Thomas Mock's blog](#)
(Customer Success Manager @ Rstudio)

R Markdown: Literate Programming

Goal: Capture code, text/comments, and output in a single document

MVP of reproducibility

- HAS to run successfully to save/knit the output
- Self-documenting (code is embedded)
- Self-contained workspace

Exploratory Data Analysis

R Markdown: As a data product

Goal: Generate output natively in R for general consumption

- Presentations (ppt, web-formats – xaringan, LaTeX formats - Beamer)
- Dashboards (flexdashboard)
- Reports (HTML, Word, PDF)
- Websites & Blogs (blogdown , distill)
- Books & Manuscripts (bookdown)

R Markdown: As control document

Goal: Scale data science tasks, automate the boring stuff, create robust pipelines

- Automation with parameters
- Child documents
- R Markdown for emails with `blastula`

R Markdown: As templating

Goal: Don't repeat yourself, generate *input* templates or *output* documents from code

- Knit with `knit::render()` which lets you generate R Markdown outputs programmatically with code
- Looping outputs
- Templating engines whisker or `usethis::use_template()`

(1) YAML = metadata

- Save output options here
- Different syntax/language than the rest of the document
- Watch out for your spaces!

```
---  
author: Your name here  
title: Your title here  
output: html_document  
---
```

```
---  
author: Your name here  
title: Your title here  
output:  
  html_document:  
    toc: true  
    toc_float: true  
  theme: flatly  
---
```

(2) Text & (3) Code

- Code chunks!
- You can think of each chunk sort of like a mini-script file within the larger document
- Text written following Markdown

syntax	becomes
Plain text End a line with two spaces to start a new paragraph.	Plain text End a line with two spaces to start a new paragraph.
italics and _italics_	<i>italics</i> and <i>italics</i>
bold and __bold__	bold and bold
superscript^2^	superscript ²
~~strikethrough~~	strikethrough
[link](www.rstudio.com)	link
# Header 1	Header 1
## Header 2	Header 2
### Header 3	Header 3
#### Header 4	Header 4
##### Header 5	Header 5
###### Header 6	Header 6



(3b) Global Setup chunk

- A special chunk label: `setup`
- Typically the first chunk
- All following chunks will use these options because it sets globally the chunk options – hence “global set up chunk”
- Set `include = FALSE` so that is is not printed out
- You can (and should) use individual chunk options too

```
9
10  ````{r setup, include=FALSE}
11  knitr::opts_chunk$set(
12    echo = TRUE,
13    warning = FALSE,
14    message = FALSE)
15  ```
16
```

Code Chunk Options

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/

Chunk Labels

GOOD

`my-plot`

`myplot`

`myplot1`

`myplot-1`

`MY-PL0T`

BAD

`my_plot`

`my plot`

`everything else!`

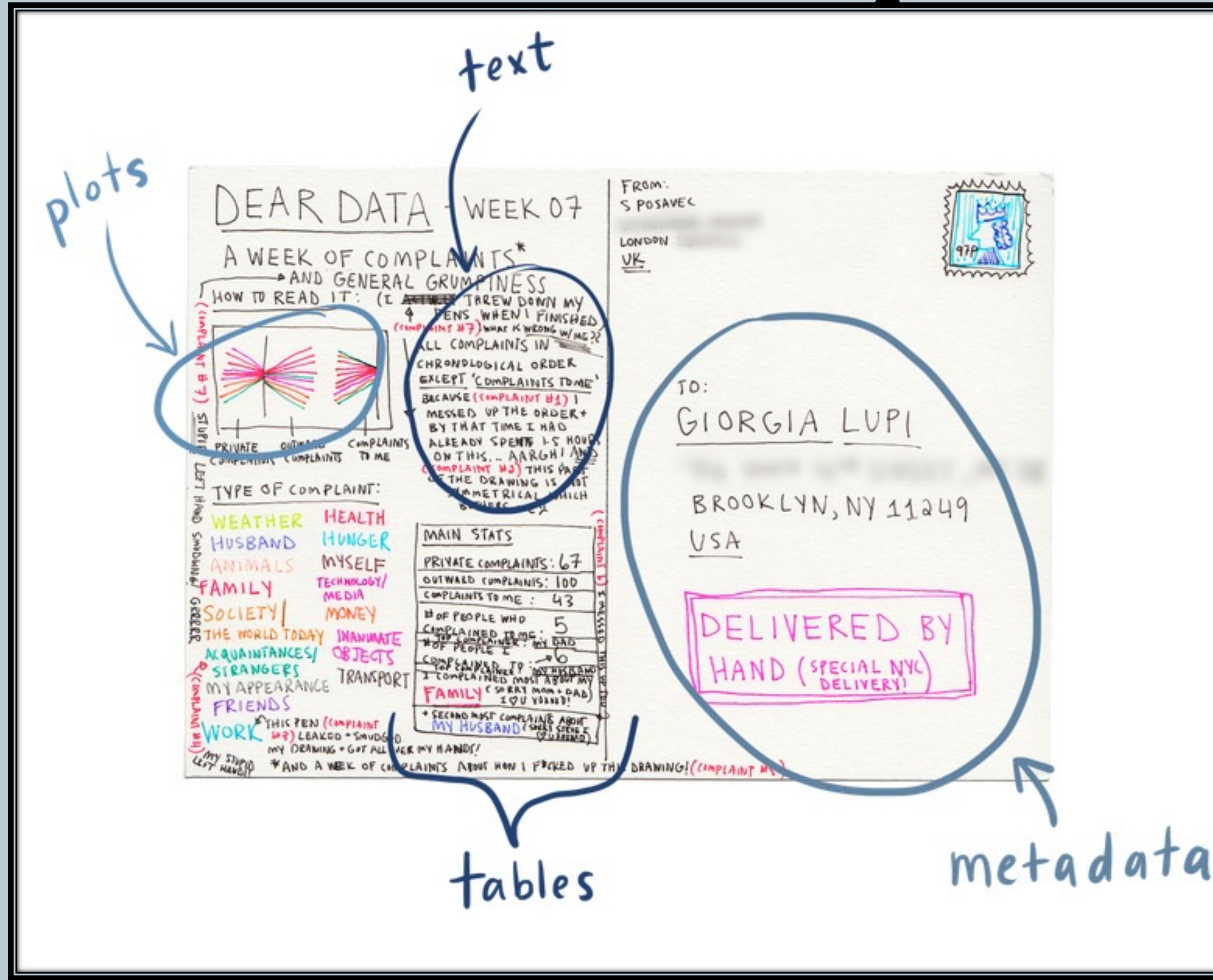
- Place between curly braces
`{r label}`
- Separate options with commas
`{r label, option1=value}`
- No duplicate chunk labels! You will get an error

The diagram illustrates the structure of an R Markdown document. The code is color-coded and annotated with curly braces and labels:

- YAML**: Annotates the YAML front matter at the top of the file.
- Text**: Annotates the introductory text and the R code chunk for `summary(cars)`.
- Text**: Annotates the explanatory text for including plots and the R code chunk for `plot(pressure)`.
- Text**: Annotates the note about the `echo = FALSE` parameter.
- Code – setup chunk**: Annotates the first R code chunk (lines 8-11).
- Code**: Annotates the second R code chunk (line 18).
- Code**: Annotates the third R code chunk (line 26).

```
1 ---  
2 title: "Title"  
3 author: "Author"  
4 date: "Date"  
5 output: html_document  
---  
|  
8 ``{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
|  
11 ## R Markdown  
14 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>.  
15  
16 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:  
17  
18 ``{r cars}  
19 summary(cars)  
|  
21 ## Including Plots  
24 You can also embed plots, for example:  
25  
26 ``{r pressure, echo=FALSE}  
27 plot(pressure)  
|  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent  
printing of the R code that generated the plot.  
31
```

RMarkdown as a postcard



Credit: Allison Hill
<https://www.apreshill.com/>

Rmarkdown

TEXT. CODE. OUTPUT.
(GET IT TOGETHER, PEOPLE.)



If you have not
done so
already, try the
following
tutorial

<https://commonmark.org/help/tutorial/>

Takes about 10 minutes



Introduction

Each lesson introduces a single Markdown concept with an example. When you see a red pulsing circle in the example, select to examine it for details.

After studying the example, try a few practice exercises with your new knowledge. Skip to any lesson at any time via the navigation controls. Experiment and have fun!

This tutorial is open source – [help us improve it!](#)

BEGIN LESSON →

WHAT IS MARKDOWN?

Intro

Emphasis

Paragraphs

Headings

Blockquotes

Lists

Links

Images

Code

Nested Lists

The End

R Studio also has a longer self-paced tutorial

<https://rmarkdown.rstudio.com/lesson-1.html>

R Markdown from R Studio®



Introduction

[How It Works](#)

[Code Chunks](#)

[Inline Code](#)

[Code Languages](#)

[Parameters](#)

[Tables](#)

[Markdown Basics](#)

[Output Formats](#)

[Notebooks](#)

[Slide Presentations](#)

[Dashboards](#)

[Websites](#)

[Interactive Documents](#)

Introduction

Overview

R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both

- save and execute code
- generate high quality reports that can be shared with an audience

R Markdown documents are fully reproducible and support dozens of static and dynamic output formats. This 1-minute video provides a quick tour of what's possible with R Markdown:



General go-to R Markdown resource

<https://bookdown.org/yihui/rmarkdown/>

By 2 of the authors of the knitr package!

The R Series

R Markdown The Definitive Guide



Yihui Xie
J. J. Allaire
Garrett Grolemund

 CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Questions?

If you have not done so already,
please fill out the mid-term
course survey!

<https://edinburgh.onlinesurveys.ac.uk/health-data-science-april-2022-mid-term-feedback>