

Introduction to literate programming with Quarto

Elizabeth Waterfield

04/09/2025

Credit statement and licence

Elizabeth Waterfield: Conceptualization, Software, Writing - Original Draft, Visualization. **Sara Lil Middleton:** Writing - Review & Editing, Supervision. **Sarah von Grebmer zu Wolfsthurn:** Conceptualisation, Writing - Review & Editing, Supervision, Project Administration, Validation.

This work by Elizabeth Waterfield, Sara Lil Middleton and Sarah von Grebmer zu Wolfsthurn is licensed under a CC-BY 4.0 [Creative Commons Attribution 4.0 International License](#).

From SvG: I think you are mixing instructor and speaker notes throughout - make sure to add what you would say for a slide to the students to speaker notes, and any tips or things instructors need to pay attention to to the instructor notes. Also make sure to have consistent formatting for these notes, like the speaker notes in bold and if there are bullet points separated by a see the first 20 slides or so and revise the rest accordingly. Also check the creation manual for naming of the images and edit!

Speaker Notes: Speaker notes may act as a guiding script for delivering the presentation. These notes contain key information, such as phrases, explanations, or transitions, that can be said aloud to clarify concepts, emphasize important points, or maintain the flow of information.

Instructor Notes: Instructor notes provide teaching support but are not intended to be communicated to the students directly. These notes contain additional context, such as learning objectives, common issues, and pedagogical tips, to help the instructor adapt their teaching to the learner's needs as well as anticipate challenges.

Prerequisites

! Important

Before completing this submodule, please carefully read about the necessary prerequisites.

Prerequisite	Description	Where to find it
Basic R skills	3.2. Introduction to R - Part I	Module 3.2.
Advanced R skills	3.3. Introduction to R - Part II	Module 3.3.
Basic Git skills	3.4. Introduction version control (Git) with RStudio	Module 3.4.
Collaboration on GitHub	3.5. Introduction to collaborative coding with GitHub	Module 3.5.
RStudio	Version 2025.05.1+513 or higher	Download Link
R	Version 4.4.3 or higher	Download Link
Zotero	Reference Management Tool	Download Link

Speaker Notes: Let us first take a look at these prerequisites. These are important to complete this submodule.

Instructor Notes: These are the prerequisites for this submodule. Before you get started on this submodule with your audience, you need to ensure that the audience fulfills these criteria. In this session, we will assume that your audience has basic R skills and completed the corresponding workshops. We also assume familiarity with Git and GitHub and that the audience has also completed those submodules. Finally, we assume that, through the previous workshops, participants have downloaded and installed R and RStudio on their local machines, and also have the desktop app for Zotero.

Questions from previous submodule?

Speaker Notes: Are there any questions from what we discussed during the last session? Are there any remaining thoughts or discussion points? **Instructor Notes:** - Aim: clarify questions from the previous submodule and/or to discuss assignments. - Additional slides may need to be added depending on the nature of the homework assignments. - It is critical for the learning process to ensure that students are on the same page and have been able

to achieve the learning goals of the previous workshop. - Not applicable if this set of slides corresponds to the first submodule of a new module.

Before we start: Survey time!

Click on the following link or scan the QR code to test your prior Quarto knowledge: [Quarto Survey](#)



Speaker Notes: - Let's start the session by gauging where our Quarto skills are at this point. Please answer honestly and don't worry about having little to no prior knowledge

about Quarto. This lesson is for beginners. **Instructor Notes:** - Aim: the pre-submodule survey serves to examine students' prior knowledge about the submodule's topic. - Use free survey software such as partify to establish this. You can use the example survey, edit it or create your own. Make sure to have a QR code for easy scanning, as well as the link displayed on the slides.

Discussion of survey results

What can we take from this?

Speaker Notes: Looking at the results from this small survey, what can we take from this?

Instructor Notes: - Aim: briefly examine the answers given to each question interactively with the group. - Use visuals from the survey to highlight specific answers. - Make it clear to the group that there will be a similar post-submodule survey to examine understanding and learning progress.

Where are we at?

Open Research is about sharing data, methods, and results to promote collaboration in science.

So, why Quarto?

- Allows for work with text, code, and media all in one place
- Makes workflows transparent and reproducible
- Integrates citation tools to manage references more efficiently
- Promotes accessibility through options for openly sharing work to others

Quarto is a tool that can help us connect ideas, data, and people through open and reproducible research.

Speaker Notes: - Let's first answer the question, "why are we learning Quarto?" Research can be messy and scattered sometimes, especially when you want to use different elements like text, code, and media. This can also be an obstacle when sharing and ensuring your work is reproducible. - This is where Quarto comes in! It brings these elements together where we can type, run code, embed media, and cite literature all in one place. It also allows for different ways to share and ensure our research is transparent and reproducible. - So this lesson is just part of a bigger journey in integrating Open Research practices in our work.

Instructor Notes: - Aim: Place the topic of the current submodule within a broader context. - Consider Quarto as a solution to some existing challenges in Open Research to remind students what they are working towards and what the bigger picture is.

What is Quarto?

An **open-source scientific and publishing system** that combines text, code and media to produce transparent and reproducible work that can be freely accessed by others.

With Quarto, you can easily:

- **Analyze** data, text, or research content
- **Share** results and outputs like reports, slides, or websites
- **Reproduce** entire workflows

 Quarto Website

You can always check out the [Quarto Website](#) to learn more.

Speaker Notes: - Quarto is an open-source publishing system, this means it is free to use and community-driven. Anyone can download it, contribute towards its improvement, and adapt it. - It's designed for combining text, code, and media in a way that makes your work transparent and reproducible, this means that anyone can read it, rerun the code, and even check the results themselves. - The 3 key things you can do with Quarto are: - Analyze and work with data, text, or research content directly in your document - Share or present your results as reports, slides, or even full websites - Make your workflow reproducible so others can repeat them. **Instructor Notes:** - Establishing a clear idea of what Quarto is at the beginning of the lesson is useful for building context and purpose - Context: how all the parts of Quarto work together to create intended document(s) - Purpose: reasons why one would use Quarto (its versatile, customizable, and shareable)

Covered in this session

1. **Key Terms and Definitions:** Understanding core concepts in Quarto
2. **Setting up Quarto:** Opening a Quarto document in RStudio
3. **Authoring:** Writing text and structuring content
4. **Code Chunks:** Running and displaying code
5. **Additional Authoring Features:** Using images, links, and columns
6. **Citations:** Adding citations and bibliography with Zotero
7. **Publishing:** Sharing your Quarto document to GitHub

Speaker Notes: - Aim: Core theoretical introduction of submodule topic. - This session will cover some basics in how to use Quarto and then some ways it can work with other platforms to perform functions like referencing and publishing. - We will first begin by

setting up a Quarto document in RStudio, then we will look at important elements: text, code, media, links, and layout. Next, we will work with both Quarto and Zotero for quick and easy citations and, finally, we will look at sharing our work to GitHub. - All of these are pertinent steps in learning how to use Quarto to facilitate Open Research practices

Instructor Notes: - Pair theoretical aspects with practical exercises and group discussions according to the Think-Pair-Share style and according to Cognitive Load Theory (Sweller, 1980). - For a 90-minute lesson, the instructor should try to “lecture” for only 20 minutes, students should work in groups/pairs/on their own for at least 55 minutes of the lesson (+ a 15 minute break). - Practical exercises on topic - Aim: practical exercises for students to apply the new skills in practise. Each submodule topic will include corresponding Tasks-adopting a “learn by doing” approach. - Depending on the topic, the exercises should be in accordance with the learning objective(s). - It’s useful to have exercises directly after a topic is taught to reinforce what was learnt. - For students who advance faster: Prepare extra exercises.

Session learning goals

At the end of this session, you should be able to:

- **Create, edit, and render** Quarto documents
- **Use** key Quarto features like code chunks, YAML headers, citations, and output formatting
- **Insert** citations and **generate** a bibliography with Zotero
- **Publish** and **share** your work using GitHub Pages

Speaker Notes: - These are the goals we’re working toward in today’s session. By the end, you should be comfortable creating, editing, and rendering Quarto documents. You’ll also know how to use key features like code chunks, YAML headers, citations, and formatting. - We’ll practice inserting citations and generating bibliographies using Zotero, and finally, we’ll publish and share our work using GitHub Pages. - Think of this as the full workflow: from authoring, through referencing, to sharing your work openly.

Instructor Notes: - Aim: Formulate specific, action-oriented goals learning goals which are measurable and observable in line with Bloom’s taxonomy (Anderson et al., 2001; Bloom et al., 1956) - Emphasis is placed on the **verbs** of the learning goals- choose verbs that align with the skills you want to develop or assess.

Key terms and definitions

- A Quarto document is saved as a **.qmd file**
- Components of a **.qmd file**:
 - **YAML header** is the section at the top of the Quarto document that controls settings like the title, output format, and author
 - **Code chunks** are the sections of the document that contain code (from R or Python, for example) that are used for showing results such as tables, plots, or calculations
 - **Quarto markdown text** combines text, codes, and formatting to create the actual content of the document

Speaker Notes: A Quarto document or file gets the extension of .qmd. There are different components within a Quarto document that are important for rendering. First, we have the YAML header, which is the section at the top of the Quarto document that controls settings like the title, output format, and author. Second, we have the code chunks, the sections of the document that contain code (from R or Python, for example) that are used for showing results such as tables, plots, or calculations. Finally, we have the Quarto markdown text, which is the component that combines text, codes, and formatting to create the actual content of the document. **Instructor Notes:** - Aim: Introduce key terms and definitions that students will come across throughout the session. - This first part of the lesson is useful to establish an understanding of important vocabulary- it can be helpful to remind students of the meaning of these terms as they appear in the upcoming sections.

“Rendering in Quarto”

What is **rendering**? The process where Quarto runs the code, combines it with the text, and creates a final output.

- There are two ways to render in Quarto:
 - **Render on Save**: Quarto will automatically re-render the document each time you click “Save”
 - **Manual Rendering**: You have to click on the “Render” each time you want to see the output



Figure 1: You can find both *Render on Save* and *Manual Render* at the top of your workspace

Speaker Notes: - Quarto documents can be rendered either manually or automatically when you save. - Manual rendering gives you control meaning you decide when to update the document, which is helpful if you're making lots of edits and don't want constant re-renders. - Render on Save automatically updates your document every time you hit Save, which is great for quick feedback and seeing your changes instantly. In this lesson, you can try both to see which workflow feels more comfortable for you.

Quarto modes

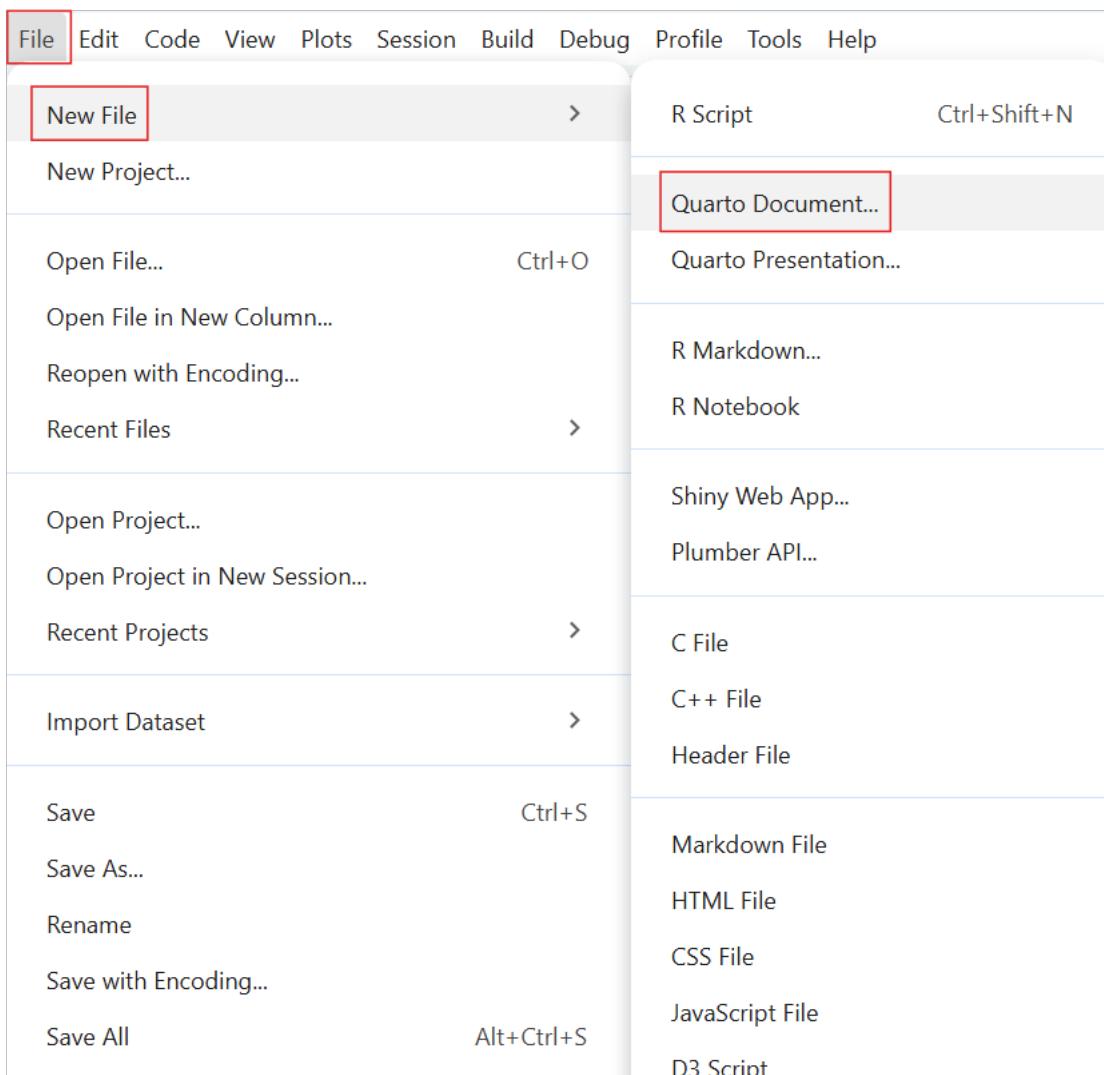
You can write Quarto documents in **Source mode** or **Visual mode** in RStudio.

- **Source mode** allows you write directly in plain text/Markdown Syntax, allowing for more control and it's closer to raw code
- **Visual mode** gives a WYSIWYM-style interface which is easier for beginners, but sometimes hides syntax.
- This is part of the **authoring process**, and it allows you to format the text, add code, and build your document.

Speaker Notes: - Base yourself on conceptual change theory and examine existing concepts in relation to some key terms. - Re-examine formation of new concepts at the end of the lesson. - WYSIWYM is an acronym for "What You Say is What You Mean" which means that whatever you see in your document will be what you see even when you save or render the document as PDF or as HTML. - Quarto documents can be created and edited in either Source mode or Visual mode. Both modes work with the same file, but they provide slightly different experiences. In this lesson, most of our tasks will be done in Source mode, but you'll also have a chance to try Visual mode so you can compare how the same content looks and feels in each. **Instructor Notes:** - "Authoring process" is mentioned here but not defined because it will be explained further in later slides.

Creating a Quarto document

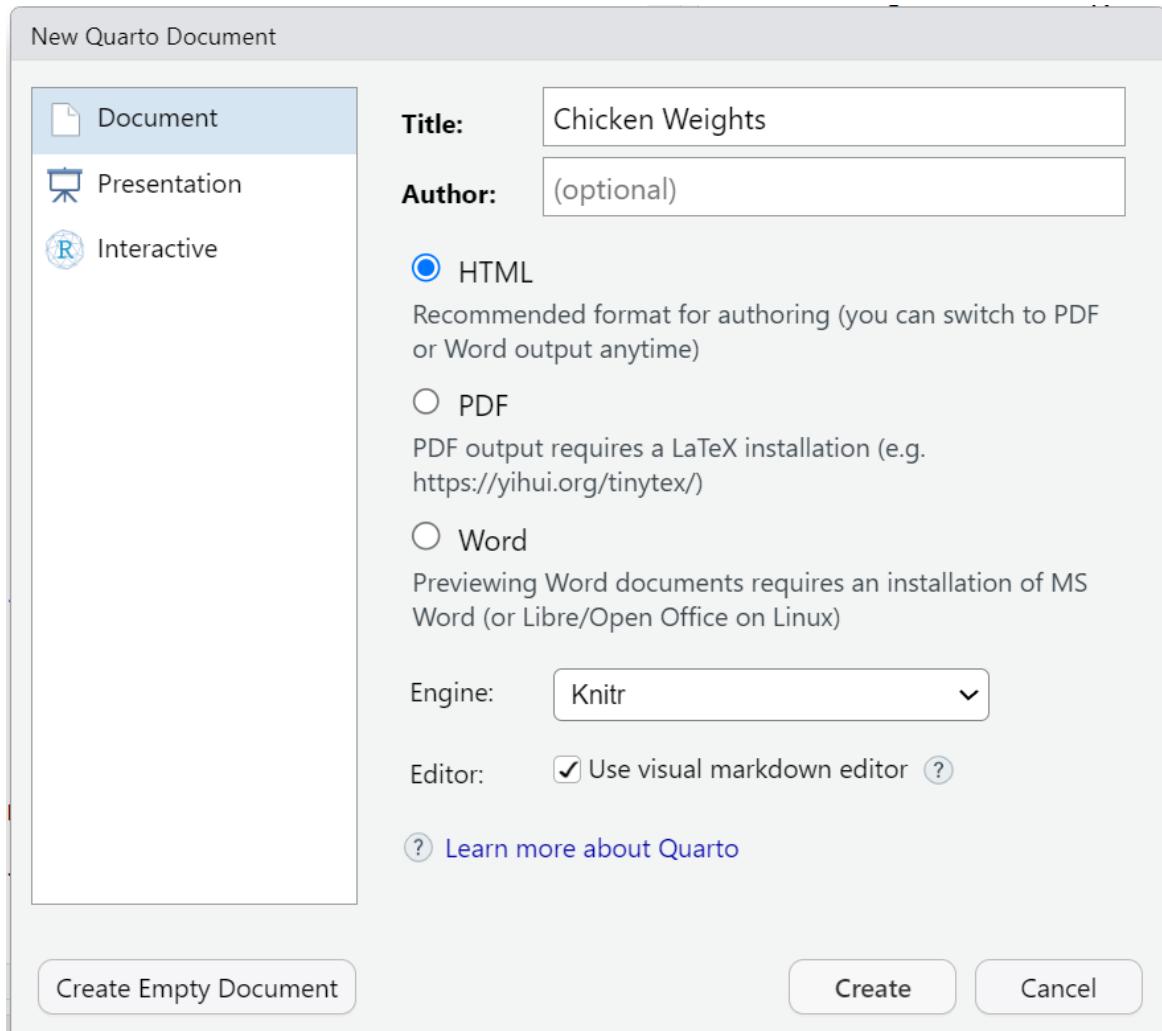
These are the steps to open a Quarto Document in RStudio:



- Select **File**
- → Select **New File**
- → Select **Quarto Document**

Speaker Notes: - To get started with Quarto in RStudio, the first step is simply creating a new Quarto document. - You see options for both Quarto Document and Quarto Presentation. What's the difference between the two? - A Quarto Document is best for reports or articles because it produces continuous text like a paper. - A Quarto Presentation is for creating slide decks, like PowerPoint or Reveal.js slides.

Setting up Quarto document



- You can set some of the YAML header details here like the title and output format
 - The default format is html
- → Hit **Create**

Speaker Notes: - This creates your document where you can now customize the YAML header, add Markdown text, and insert code chunks. - The default format is html but this can be changed later on by editing the YAML header directly in the document.

Practical exercise 1

- Repeat the steps to create a new Quarto Document

Speaker Notes: ADDD **Instructor Notes:** Ensure students know where their Quarto project folder is located on their device. This will make it easier for them to find files in later tasks. Encourage them to save the document right after creating it by selecting “Save As”, so they can intentionally choose the document location (for example, on the Desktop or a dedicated course folder).

Authoring a Quarto document

What is **authoring**? The process of writing and structuring the Quarto document.

- Consider it as a formula: **Authoring = YAML Header + Markdown**
- To practice authoring in Quarto, let's start with setting the YAML header and adding markdown text in **Source mode**.

Speaker Notes: - Authoring is combining the YAML header with the markdown text. This means that what you do to customize, add to, and develop your document is the process of authoring - The YAML header is the section at the very top, wrapped in three dashes (—). It holds the document's settings, like the title, author, date, and output format. - Below that, we use Markdown text to write the actual content which are things like headings, bullet points, bold or italic text, and the actual body of your document. **Instructor Notes:** - Reminding students on the meanings of YAML Header and Markdown text will be helpful especially to those with very little prior knowledge as these are terms that appear very often when working with Quarto.

Practical exercise 2

- Copy and paste the following into the YAML Header of your document

```
---
```

```
title: "ChickWeight Analysis"
author: "Your Name"
format:
  html:
    code-fold: false
    toc: true
---
```

Output format

You can replace “html” to render the document to a different format.

Here is a link to a list of the different output formats.

Speaker Notes: - Remember that the YAML Header goes at the top of the document in both Source and Visual Modes. - In the YAML header, the format field tells Quarto what kind of output to create. Here we’ve set it to html, which means the document will render as a web page. You could choose other formats like PDF or Word, but for this lesson we’ll stick with HTML so everyone has the same experience.

Instructor Notes: There is a small clipboard icon at the top-right of the text box. Let students know they can simply click the icon to copy all the text in that specific text box. This icon will appear for every text box moving forward so you can remind them of it the next few times a task requires them to copy the content of a text box.

Authoring: Text formatting

Basic Markdown Text Formatting

- **Bold:** `**bold**` → bold
- **Italic:** `*italic*` → italic
- **Strikethrough:** `~~text~~` → text
- **Inline code:** ``code`` → code

Markdown text shortcuts

Here is a link for the full list of markdown text shortcuts for formatting!

Speaker Notes: - With Markdown text, we can format text with very simple symbols. It works the same way in both Source and Visual mode with the only difference being how you see it while typing. - In Source mode you see the symbols, while in Visual mode it looks like regular bold or italic text right away.

Authoring: Markdown text

Have you ever wondered what affects a chick's weight?
This document explores the ChickWeight dataset using R.
The goal is to compare chick weight across different diets and time points.
Key steps include:

- Loading the dataset
- Visualizing growth trends
- Summarizing results

This is an example of **markdown text**. Let's use it to continue authoring and building our document.

Speaker Notes: - Aim: Help students feel comfortable using Markdown text by showing how simple and familiar it really is. - the words “Markdown Text” may seem technical or complicated for those new to Quarto. But Markdown text can simply be understood as just typing (like writing in any normal document), sometimes with a few simple symbols to add structure or emphasis.

Practical exercise 3

- Copy the markdown text from the previous slide and paste it to your document
- Bold the word “*ChickWeight*”
- Italicize the phrase “*growth trends*”

Speaker Notes: ADD **Instructor Notes:** Instruct students to render their document after each practical exercise (where applicable) so they can see the changes as they make them and identify issues.

Authoring: Callout boxes

Highlighting particularly important aspects with **Quarto callout boxes**

! Important with Title

This is an example of a callout box to highlight particularly important information using `callout-important`

Tip with Title

This is an example of a callout box to give important tips using `callout-tip`

Note with Title

This is an example of a callout box to include an additional note using `callout-note`

Speaker Notes: - Callout boxes are a way to highlight important information in your document. They stand out visually, so readers immediately notice them. You can use them to emphasize tips, warnings, examples, or key takeaways. - Acknowledge that the look of the callout boxes changes depending on if “-important,” “-tip,” or “-note” follows the word “callout” **Instructor Notes:** These are just examples of callout boxes. How to actually create one will be covered in the next slide/task.

Authoring: Inserting callout boxes

Here is the markdown text for inserting a **callout note** box:

```
::: callout-note  
## Based on Real Data  
  
The ChickWeight dataset in R is based on real experimental data.  
:::
```

Speaker Notes: - Inside the box, you can add a heading and text. - These aren't just for highlighting important information but can be used for supplementary or by-the-way information as well as warnings.

Practical exercise 4

- Copy and paste the markdown text from the previous slide into your Quarto document to add this callout note box.

Speaker Notes: - The details in the formatting are very important (for callout boxes and for all authoring features) - Specific details: - beginning and ending in 3 ::::; - single space between the first :::: and callout-note; - single space between ## and title

Instructor Notes: - Aim: Understand and practice the specific format of a callout box. - When rendered, acknowledge how the callout boxes sets its contents apart from the rest of the document to draw the reader's attention

Code chunks

There are two ways to insert **code chunks**:

- Manually type 3 back ticks `` then {r} to start a coding chunk, enter your code, then end it with 3 back ticks

OR

- Use the keyboard shortcut Ctrl + Alt + I (Windows/Linux) or Cmd + Option + I (Mac) to insert a code chunk then simply enter your code

Speaker Notes: - Code chunks are sections of a Quarto document where we can run code directly inside our file. They're marked by three backticks followed by the language name, like {r} or {python}. - These let us write and execute code, and then display the results, such as tables, plots, or calculations, right in the document. Code chunks make it possible to combine text and analysis in one place, so the document stays reproducible and dynamic.
Instructor Notes: - using {r} to start the Code Chunk indicates that the coding language that we're using is R. This is relevant for this lesson as students will be using R Studio and possession of some background knowledge in R is assumed.

Inserting code chunks

```
summary(ChickWeight)
```

```
library(ggplot2)
ggplot(ChickWeight, aes(x = Time, y = weight, color = Diet)) +
  geom_line(aes(group = Chick)) +
  labs(title = "Chick Growth Over Time")
```

These are examples of **R code**. Let's use it to insert code chunks in our document.

Speaker Notes: - For example, here we're summarizing the ChickWeight dataset, and below, we're using ggplot2 to create a visualization of chick growth over time. Notice how the code is written inside a chunk, and when rendered, Quarto automatically runs the code

and shows the results in your output. This is what makes Quarto so powerful, it combines text, code, and results all in one place.

Instructor Notes: - It's also possible to see the output of the code before rendering by clicking the small, green arrow at the top-right corner of the code chunk. Hover over it and it would say "Run Current Chunk."

Practical exercise 5

- Insert the code from the previous slide in *two separate code chunks*

Speaker Notes: - ADD **Instructor Notes:** - Students don't need to hit Ctrl + Enter (Windows) or Command + Enter (Mac) to run the code. After placing it in the coding chunk, rendering the document will run the code and print the output. - Encourage students to try using both methods to insert coding chunks for this exercise. - Note that the code is in two separate text boxes/coding chunks for teaching purposes but can actually both be put in 1 coding chunk. You do not have to start a new coding chunk each time you want to use a different R function. - After students render the output and observe how the code appears together with the markdown text from the previous activities, use this as an opportunity to exemplify and emphasize how Quarto is useful for combining text and code into one document.

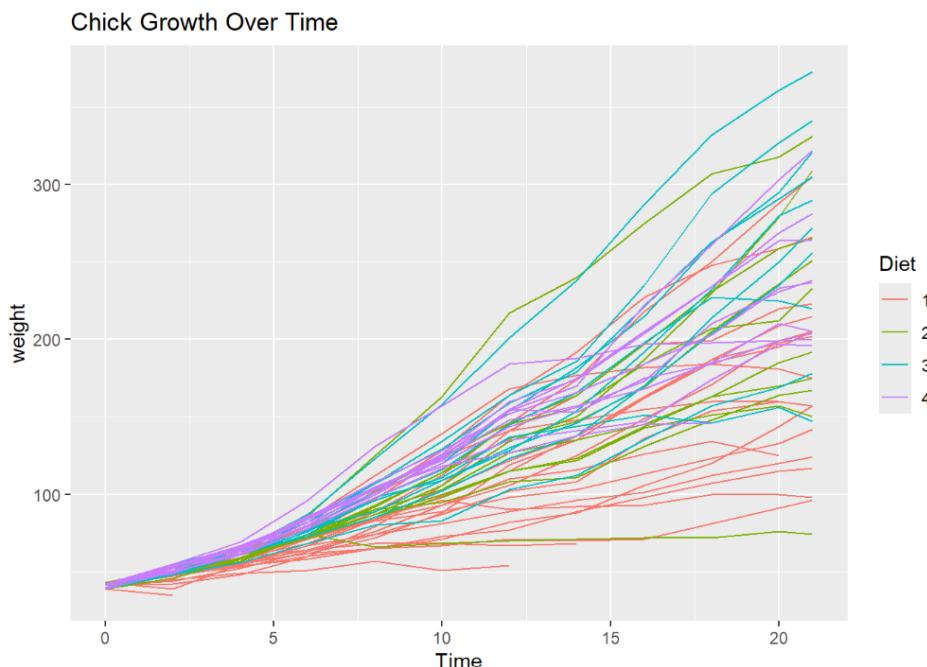
Adjusting the code chunks

Adjust how the code is portrayed by editing the YAML header:

```
summary(ChickWeight)
```

	weight	Time	Chick	Diet
Min.	: 35.0	Min. : 0.00	13 : 12	1:220
1st Qu.	: 63.0	1st Qu.: 4.00	9 : 12	2:120
Median	:103.0	Median :10.00	20 : 12	3:120
Mean	:121.8	Mean :10.72	10 : 12	4:118
3rd Qu.	:163.8	3rd Qu.:16.00	17 : 12	
Max.	:373.0	Max. :21.00	19 : 12	
				(Other):506

```
library(ggplot2)
ggplot(ChickWeight, aes(x = Time, y = weight, color = Diet)) +
  geom_line(aes(group = Chick)) +
  labs(title = "Chick Growth Over Time")
```



- Currently in our YAML header, we have it set to `code-fold: false` which means the code is visible and not collapsible.

Speaker Notes: ADD **Instructor Notes:** - Clarify that “not collapsible” means that we cannot choose to hide the code, we cannot fold it closed (as we will see when we set it to `code-fold: true`) - Quarto gives us several options for controlling how code appears in our documents. For example, we can choose whether to show or hide the code itself, whether readers can fold code open and closed, or whether only the results are shown. - These settings don’t change the analysis, but just change how it’s displayed. This flexibility is helpful because sometimes we want to highlight the process by showing the code, and other

times we want readers to focus on the results. - In open research, showing the underlying code isn't just a technical choice, it's important for building trust and reproducibility.

Code chunks display settings

Here are some other code display settings that we can incorporate into our YAML header:

- `code-fold: true` collapses the code so the reader can expand it
- `code-tools: true` adds the functions “show code” at the top of the page and “copy” next to the chunks
- `echo: true` both the code and the output is visible

Let's edit the YAML header to make the code chunks **collapsible** and add **code tools**.

Speaker Notes: - `echo: true` and `code-fold: false` both affect how code is shown, but in different ways. `echo` decides if the code prints at all, while `code-fold` lets readers toggle it open or closed, deciding whether the code can be hidden or expanded. **Instructor Notes:** - It's easy to confuse `echo` and `code-fold` because they both deal with code visibility. Be clear on the difference.

Practical exercise 6

In the YAML header:

- Change `code-fold: false` to `code-fold: true`
- Add `code-tools: true`

Speaker Notes: ADD **Instructor Notes:** Engage students by asking for possible reasons why one would want their code to be visible or not.

Pre-break survey

Speaker Notes: ADD **Instructor Notes:** - Before the break, take a moment to review the things we covered in the first part of this lesson. It's important to have an idea of these key terms because they will reappear many times throughout the rest of this lesson.

What's the name of Quarto Markdown document?

- a. .png file
- b. .qmd file
- c. .doc file
- d. .mp3 file

Speaker Notes: This survey has a few simple recap questions about what we've covered so far in Quarto. The goal isn't to test you, but to help you reflect on what you've learned and for me to see if anything needs a bit more clarification. **Instructor Notes:** - Aim: This pre-break survey serves to examine students' current understanding of key concepts of the submodule - Use free survey software such as or other survey software (Menti, participy, formR) to establish the following questions.

What is the YAML header?

- a. summarizes the document into a single line
 - b. this is where you store notes and reminders
 - c. it's like the "settings" of the document
 - d. just the title of the document
-

Which of the following is used to format content (like paragraphs and bullet points)?

- a. Code chunks
- b. YAML header
- c. Markdown text
- d. All of the above

What are the two components of the Authoring process?

From SvG: Are there answers to this question and the one on the next slide?

How to create a code chunk in a Quarto Document?

Break! 5 minutes

Post-break survey discussion

Speaker Notes - Aim: To clarify concepts and aspects that are not yet understood - Highlight specific answers given during the survey - Also, use this time to clarify any confusions or questions on specific topics students may have from the first part of the session.

Additional authoring features

Quarto offers additional **authoring features** that make it more versatile and comprehensive. These include:

- Adding **links** and **hyperlinking** text
- Embedding **media**
- Creating **multi-column layouts** and more.

i Commonly used authoring features

Here is a link for commonly used markdown syntax for additional authoring features.

Speaker Notes - Beyond writing plain text, Quarto also allows us to add features like hyperlinks, images or videos, and multi-column layouts. These improve how our document communicates. - Hyperlinks let us connect to sources or other sections, media like images and video help explain complex ideas more clearly, and multi-column layouts make the content easier to read and more visually appealing. - Using these features thoughtfully improves both the usability and the overall look of the document.

Inserting links

Link with title: [title] (link)

- [Quarto Website] (<https://quarto.org>) → [Quarto Website](#)
- [Click here to see more information on reveal.js] (<https://revealjs.com>)
→ [Click here to see more information on reveal.js](#)

Link without title: <https://link>

- <<https://www.markdownguide.org/>> → <https://www.markdownguide.org/>
- <<https://github.com/>> → <https://github.com/>

Speaker Notes - You can insert a link with a title using square brackets for the text and parentheses for the URL. This is helpful when you want the link to be descriptive or if you want to hyperlink written text. - You can also add a link without a title by simply wrapping the URL in angle brackets (<...>). This shows the raw link, which is fine if the URL itself is clear and concise enough. - A good rule of thumb is to use titled links for readability, and untitled links when you want to show the exact URL.

Inserting images

- Image by itself with no caption: ![] (path/image.png)



Speaker Notes - A benefit of using Quarto is the ability to easily add media, such as images, to your documents. - For this lesson, we will focus on just adding images. You can add the image by itself with no additional text, such as a caption. - The “path” tells Quarto where to find the image on your device. If the image is stored in the same folder as your .qmd file, you only need to use the image’s file name. If the image is inside a subfolder (For example, an images folder next to your .qmd), then you just include the folder name in the path, like images/picture.png.

Inserting images

- Image with a caption below it: ! [caption] (path/image.png)



Figure 2: This is a caption about three yellow chicks in the grass.

Speaker Notes - Images can also be inserted with a caption placed below it. - Helps with clarity and accessibility, since captions explain the relevance of the image. - Also useful in reports, presentations, or academic writing where figures must be labeled.

Inserting images

- Image that redirects you to a link: ! [caption] (path/image.png)] (link)



Figure 3: Click on the image to see a study on broiler chicks

Speaker Notes - Images can also be inserted with links embedded in them. - This allows readers to click and explore more details (for example, linking a screenshot to a website or dataset). - Keeps documents concise while still offering extra resources for those who want them.

Practical exercise 7

- Click this link to download the image
- Save the image to the main folder of the Quarto document files

! [No chicks were harmed in the making of this document.] (images/chickpic1.png)

- Copy and paste the markdown text above to insert an image with a caption (edit path if needed)

Speaker Notes - Aims: to reinforce how to add images, how to insert captions, and how paths work. - Simply click the link and the image will automatically download. - The text inside the square brackets is the caption, and the text inside the parentheses is the path to the image file. - Reminder that the path tells Quarto where to look for the image on your device. - If your image is saved directly in the same folder as your .qmd file, you only need the file name. If it's in a subfolder, include the folder name in the path. *Instructor Notes* - Setting the path can be tricky depending on where the files are located - It's a good practice to save these files in the same main folder as your Quarto document files (ideally in a dedicated subfolder named "images" to keep paths organized and easy to manage). If students saved and placed their main folder earlier, then locating it should be easy. - in the Task above, the text says "images/chickpic1.png" assuming the picture is saved in a folder called "Images" in a main folder containing all the Quarto document files.

Creating columns on the slide

You can change the layout of a section using **columns**:

- Start the column layout by writing “::::: columns”
- → Begin the first column with “::: column”
- → Add content and end it with “:::”
- → Begin second column with “::: column”
- → Add content and end it with “:::”
- → End section with a final “:::::”

i List of page layout options

Here is a link for page layout options to author how content looks on your document.

Speaker Notes - Columns are useful when you want to organize content side by side instead of in a long vertical flow. For example, you might place an image in one column and an explanation in the other, or compare two code outputs directly next to each other like comparing two separate plots. - Columns help make information easier to scan and can improve the balance and readability of your slides or documents.

Practical exercise 8

Side-by-side image and bullet points with columns

Let's use columns to add bullet points next to an image.

- Click this link to download another image
- Save the image to the main folder of the Quarto document files

Speaker Notes - One of the ways you can utilize columns is by placing an image on one side and bullet points next to it. - The link leads to separate image from the previous one, please click to initiate another download. - Reminder that where the image is stored is important for setting the path.

Practical exercise 8

```
::::::: columns
::: column

! [] (images/chickpic2.png)

:::
::: column

- Monitor growth trends over time
- Compare diets and weight gain

:::
:::::::
```

- Copy and paste the markdown text to add an image with bullet points using columns (edit path if needed)

Speaker Notes - The image in this exercise will be inserted without caption, hence, the square brackets are left blank. But it is still necessary to add them. - This is also an opportunity to see how bullet points are added: by using a dash (-), followed by a single space, then the content. *Instructor Note* - The details of the syntax for columns is very important here too: - beginning and ending with exactly 6 ::::::: - beginning the layout with the word `columns` (plural) then using `column` (singular) to divide each subsequent column - a single space between ::::::: and `columns` as well as between :::: and `column` - Mistakes here will result in warning messages when rendering

Adding citations

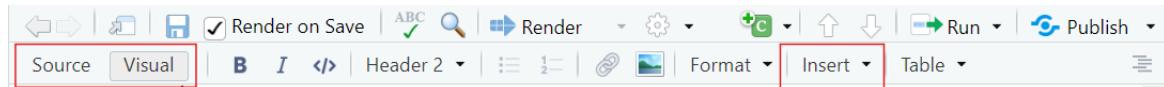


What is **Zotero**? This is a free reference management tool to collect, organize, cite, and share research sources.

- You can use **Zotero** in RStudio to easily insert citations into your Quarto document.

Speaker Notes - Zotero is a reference manager that helps you collect and organize research articles, books, and other sources. When you connect Zotero to Quarto, you can easily insert citations while writing and then automatically generate a reference list at the end. This saves time and reduces errors compared to typing citations manually.

Adding citations with Zotero



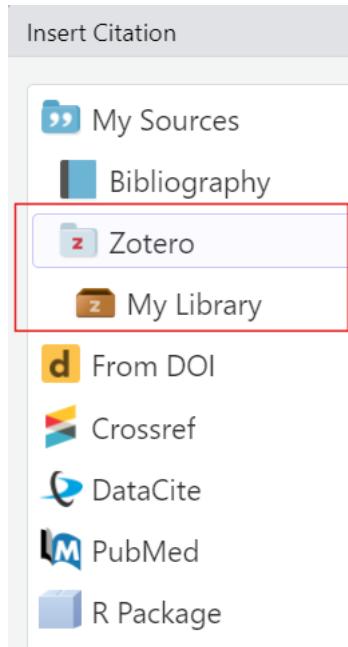
To insert a citation:

- Open Zotero on your device
- → Switch to the “Visual Editor” mode
- → Add `bibliography: references.bib` in the YAML header
- → Press the “Insert” button in the toolbar and select “Citation”

A reference list or bibliography is automatically generated at the end of the document with all the citations used.

Speaker Notes - To add citations in Quarto, you'll need Zotero open on your device. - In RStudio, switch to the Visual Editor, which makes citation insertion easier. - Make sure your YAML header includes `bibliography: references.bib` so Quarto knows where to look for your references. - Then, simply go to the toolbar, click Insert → Citation, and you'll be able to search your Zotero library directly. - Once you add citations in the text, Quarto automatically generates a reference list or bibliography at the end of your document so that you don't have to format it manually.

Adding citations with Zotero



- → Select citations straight from the folders in your **Zotero library**
- Each citation is assigned a short citation key based on the author and year (for example, “pauwels2015”).
- You can quickly cite it in your document by typing @ followed by the citation key.

Speaker Notes - When you click “My Library,” it mirrors what folders you’ve created in Zotero Library. This helps you navigate to the folder where the citation is stored. - After inserting a citation once, you see what citation key is assigned to it for easier referencing later in the text. But the way the citation key is written is different for regular and in-text citations. *Instructor Note* - the Zotero program needs to be open on the device for the “Zotero” option to show up in RStudio like it does in the image.

Practical exercise 9

Let’s practice adding a citation using **Zotero**.

- Open Zotero on your device
- Add `bibliography: references.bib` to your YAML header
- [Follow this link to an article on diets and broiler chicks](#)
- Add the citation to your Zotero Library

Diet affects chick body weight. In this study, low-energy feed made fast-growing chicks light

- Copy and paste the text above and place the cursor at the end of the sentence

Speaker Notes - Aim: to show how using Zotero with RStudio to insert citations and generates bibliography simplifies the referencing process - Follow these steps to add a citation to your Zotero library and add some information extracted from it to your Quarto document.

Instructor Notes - There are several ways to add a citation to your Zotero library but an easy way is using the ‘magic wand’ (adding the item by its identifier). Here, you can just copy the DOI of the study, hit Enter, and it will be added to Zotero. The DOIs of the articles in these tasks will be clearly visible so this method is recommended.

Practical exercise 9

- In **Visual Mode**, click “Insert” and find the article in the Zotero Library
- Click “Insert” to add the citation

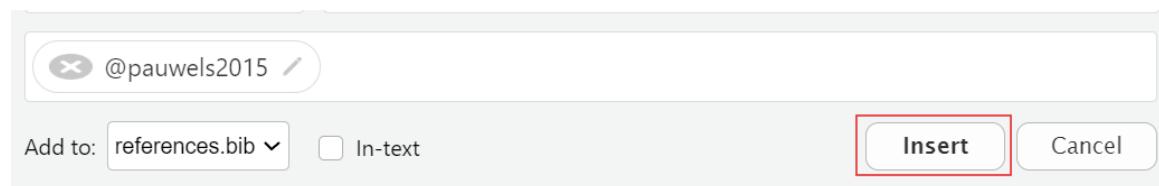


Figure 4: When you find the article in the Zotero Library folder here, select it, then click Insert

Speaker Notes - Follow these steps to add the citation to the end of the text. - It is possible to add citations in Source mode, but it is easier and more efficient to do it in Visual mode.

Instructor Notes - This is also a good opportunity for students to see all the work they’ve done so far in the Quarto document now in Visual mode.

Practical exercise 10

Let’s practice adding **in-text** citations:

- [Follow this link to another article on diets and broiler chicks](#)
- Add the citation to your Zotero Library

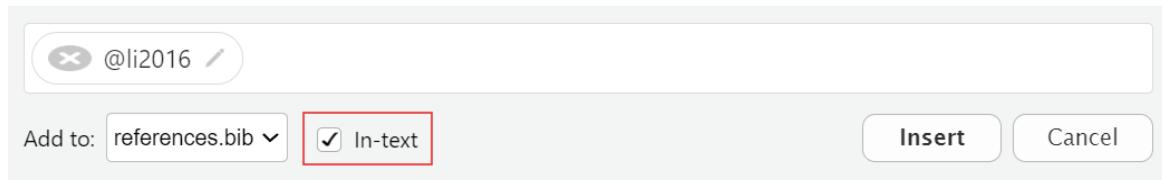
found that broiler chicks fed a higher nutrient-density diet gained more weight and did so

- Copy the text above and place the cursor at the beginning of the sentence

Speaker Notes - Follow these steps to place an in-text citation at the beginning of the text. This is another article, meaning that you will have to click the link to go to it and add it to your Zotero library. - Reminder that in-text citation is a short reference within the sentence itself that points to the full source that you're citing.

Practical exercise 10

- Click “Insert” and find the article in the Zotero Library
- Check the box next to “In-text”



- Click “Insert” to add the citation

Speaker Notes - This box needs to be checked and un-checked accordingly when adding citations.

Adding citations with Zotero

Now you should have

- A citation
- An in-text citation
- A reference list (bibliography)

Diet affects chick body weight. In this study, low-energy feed made fast-growing chicks lighter, while slower-growing chicks compensated by eating more ([Pauwels et al. 2015](#)).

Li et al. (2016) found that broiler chicks fed a higher nutrient-density diet gained more weight and did so more efficiently.

References

Li, Jianhui, Jianmin Yuan, Zhiqiang Miao, Zhigang Song, Yu Yang, Wenxia Tian, and Yuming Guo. 2016. "Effect of Dietary Nutrient Density on Small Intestinal Phosphate Transport and Bone Mineralization of Broilers During the Growing Period." Edited by Carlos M. Isales. *PLOS ONE* 11 (4): e0153859.

<https://doi.org/10.1371/journal.pone.0153859>.

Pauwels, Jana, Frank Coopman, An Cools, Joris Michiels, Dirk Fremaut, Stefaan De Smet, and Geert P. J. Janssens. 2015. "Selection for Growth Performance in Broiler Chickens Associates with Less Diet Flexibility." Edited by John Parkinson. *PLOS ONE* 10 (6): e0127819. <https://doi.org/10.1371/journal.pone.0127819>.

Changing the citation style

You can change the citation style by using csl files directly from the Zotero website. For example, putting “csl: <https://www.zotero.org/styles/apa>” in your YAML changes the citation style to APA.

Speaker Notes - When you render, you should have three key elements in your Quarto document: a citation, an in-text citation, and a reference list or bibliography at the end. - As you can see, Quarto does most of the heavy lifting as long as you insert citations correctly and the bibliography builds automatically. - Also, if you ever need to follow a specific format, like APA or Chicago, you can easily change the citation style by adding a CSL file in your YAML. It can be a link directly to the Zotero website so that you don't have to download any additional files.

Publishing

What is **publishing** with Quarto? The process of sharing the rendered documents or projects online so it can become accessible to others.

- You can publish to:
 - GitHub Pages
 - Quarto Pub
 - A personal website

i List of publishing services

Here is a link for a list of publishing services and more information!

Speaker Notes - Publishing in Quarto is all about making your work accessible to others. Once you've created and rendered your document, you can share it online so that collaborators, students, or the wider community can view it. Quarto supports multiple publishing options, including GitHub Pages, Quarto Pub, or even your own personal website.

Instructor Notes - The key takeaway here is that publishing turns your local work into something that can be accessed from anywhere.

Publishing

When sharing your work, it's important to follow established **rules and standards**. This includes:

- Considering the legal aspects
- Applying appropriate licenses
- Proper archiving

i List of publishing services

Click here for the LMU OSC page on all you need to know about sharing, copyright, best practices and more.

Speaker Notes - As practitioners of open research, sharing your work is more than just making it available. It's also about being responsible and following certain standards. There are legal aspects of sharing that we should consider such as - Licenses that define how other may use your work, - The importance of archiving to preserve your work in the long run - Established best practices to make sure our research remains credible and reusable.

Instructor Notes - Aim: To inform students that open science involves thoughtful practices. In this case, sharing should be done with legality, licenses, and reusability in mind. - We will not practice this in the session but it is good to know moving forward.

Publishing to GitHub

A great option to share your document is **publishing to GitHub**.

- **GitHub** is a platform for hosting and sharing code and projects online
- An advantage of GitHub is it allows for **version control**- the ability to track and manage changes over time
- A free and widely used way to publish your Quarto document to GitHub is by using **GitHub Pages**

Speaker Notes - GitHub is one of the most common platforms for sharing code and documents. One of its biggest advantages is version control, which means you can track every change made to your files and even go back to earlier versions if needed. - For Quarto, publishing to GitHub is a great option because it's free, widely used, and makes your work easily accessible to others. - The simplest way is through GitHub Pages, which lets you turn your Quarto document into a live website that anyone can access.

Publishing to GitHub

GitHub Pages enables you to publish content based on source code managed within a GitHub repository.

- **3 methods** to publish a Quarto document to GitHub Pages:
 1. Render to the `docs` directory and checking it into your repository
 2. Use the `quarto publish` command
 3. Use GitHub Actions to auto-render and publish whenever you push changes

Speaker Notes - GitHub Pages makes it possible to publish Quarto content as a live website, directly from your GitHub repository. There are three main ways to do this: - The simplest way is to render your document to a `docs` folder and commit it to your repo. GitHub Pages can then serve those files as a website. “Docs” here is a folder in your project where you put the files that should be published as a website. - You can also use the `quarto publish` command, which pushes your rendered site to GitHub Pages automatically. - For a more advanced setup, you can use GitHub Actions so that every time you push changes to your repo, the site automatically re-renders and updates online. - Whichever method you use depends on how automated you want the process to be.

Publishing to GitHub

Let's publish our document using the first method:

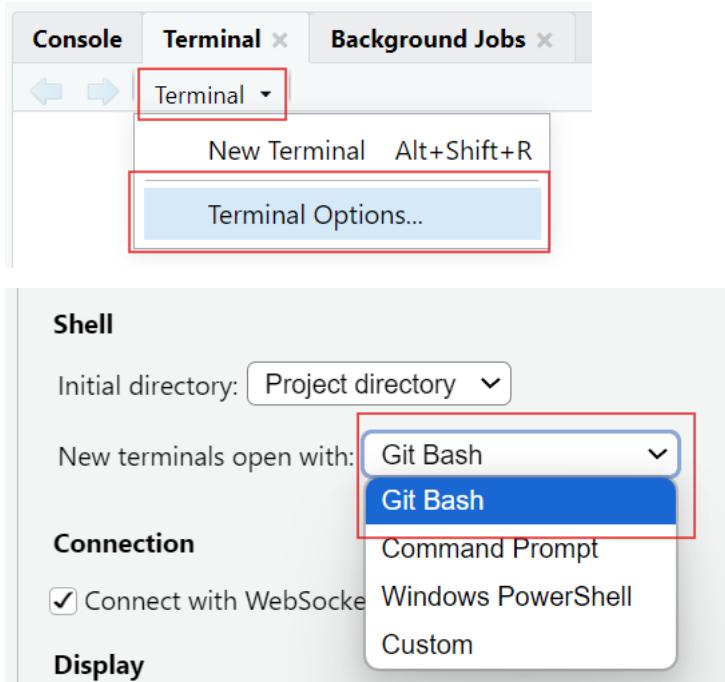
Render to docs

Speaker Notes - For this lesson, we will publish our completed Quarto document with all the additions we've made to Git Pages using the "Render to docs" method.

Publishing to GitHub

Work with **Git Bash** directly from the **Terminal** in RStudio.

- click Terminal → Terminal Options and set it to open with Git Bash



- then, click Terminal → New Terminal to begin working with Git Bash

Speaker Notes - When publishing to GitHub, we'll often use Git Bash, which is a command-line tool for running Git commands. Fortunately, you don't have to leave RStudio to use it by opening Git Bash directly inside RStudio's Terminal tab. - To do this, go to Terminal → Terminal Options, and under 'New terminals open with,' select Git Bash. Then, when you click Terminal → New Terminal, it will open Git Bash automatically, ready for you to start working. - This setup keeps everything in one place so you don't need to switch between RStudio and an external terminal. *Instructor Notes* - Git Bash should be installed on the

student's device. - When you click the drop-down arrow by Terminal, the options may look different than what's in the image in this slide, but the "Terminal Options..." will be at the end (you may have to scroll a bit). - The last step (open a new Terminal) is important because the current Terminal does not automatically change to Git Bash.

Practical exercise 11

- In your Quarto Document, edit YAML and add:

```
project:  
  type: website  
  output-dir: docs
```

- Set Terminal to open with Git Bash and open new Terminal
- Change directory to the path of the main folder with the Quarto files using `cd`

Speaker Notes - In this exercise, we're preparing our Quarto document so it can be published on GitHub Pages. First, edit the YAML header of your Quarto document to define it as a website project and set the output to the docs folder. This tells Quarto where to place the rendered files that GitHub Pages will serve. - Next, set up your terminal in RStudio to use Git Bash by following the steps in the previous slide. - From there, you'll use the `cd` command to change into the main folder where your Quarto files are stored. This step makes sure any Git commands you run apply to the right project folder. *Instructor Notes* - Reminder that copying and pasting with keyboard shortcuts do not work the same in Git Bash. Right-clicking and selecting Paste is an easy option. - Sometimes in Git Bash/Terminal, just copying and pasting will run the command, but sometimes you need to hit Enter. - Setting the directory can be tricky. - Backslashes (\) need to be changes to forward slashes (/). - If you're pasting the exact path of the folder, wrap it in quotation marks ("...") - OR an easy way so change the directory is by typing `cd` then selecting and dragging the folder directly into Git Bash- this puts the folder path after the `cd`

Practical exercise 11

```
echo "project:  
  type: website  
  output-dir: docs" > _quarto.yml
```

- Make a simple `_quarto.yml` file by copying and pasting the above in the Terminal

```
touch .nojekyll
```

- Copy and paste the above into the Terminal to add a `.nojekyll` file

i Why we add `.nojekyll`

Adding a `.nojekyll` file to the root of your repo tells GitHub Pages not to do additional processing of your published site using Jekyll

Speaker Notes - Here, we're setting up the basic project files GitHub Pages needs in order to publish our Quarto website. - First, we'll create a `_quarto.yml` file using the Terminal. This file tells Quarto that the project is a website and that the rendered output should go into the `docs` folder. - Next, we'll add a special file called `.nojekyll`. By default, GitHub Pages uses Jekyll to process websites, but that can interfere with Quarto's output. Adding a `.nojekyll` file tells GitHub Pages to leave our site exactly as Quarto created it.

Practical exercise 11

```
quarto render
```

- Copy and paste the above into Terminal to generate HTML in the `docs/` folder

```
git init  
git branch -M main
```

- Copy and paste the above into Terminal to initialize Git and rename “master” to “main”

Speaker Notes - Running `quarto render` in the Terminal takes your Quarto project and produces the HTML files in the `docs` folder. These are the files GitHub Pages will serve as your website. - Next, we need to initialize Git in the project folder so we can connect it to GitHub. The command `git init` sets up Git tracking, and `git branch -M main` renames the default branch from “master” to “main,” which is the standard branch name used in this session. *Instructor Notes* - You will see a warning about `references.bib` not found in `resource path` which usually happens because Quarto looks for the file in the project’s root or resource path, but Git Bash is case-sensitive and strict with paths. - So, you will see in your completed Git Pages web page that the citations look strange but we will leave it like that for now. - Extra steps are needed to correct this that will be covered in the Homework Assignment/Extra Activities PDF.

Practical exercise 11

- Go to [GitHub](#) and create a new repo
- In Terminal, connect to local project by using “`git remote add origin`” + the repo’s HTTPS URL (for example, “`git remote add origin https://github.com/yourname/chickweight-ex`

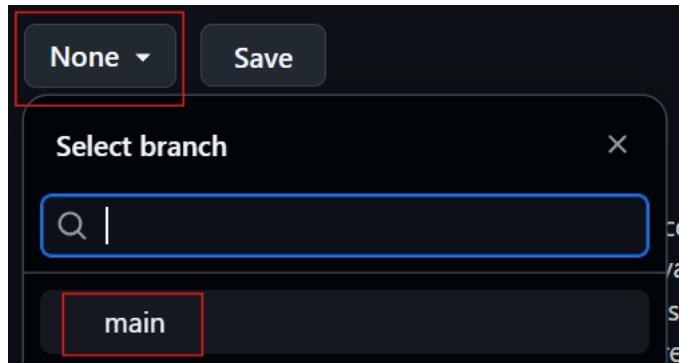
```
quarto render  
git add docs  
git commit -m "Publish site to docs/"  
git push -u origin main
```

- Copy and paste the above into Terminal to render your site and push it to GitHub
- In your repo on [GitHub](#), go to Settings
- Click “Pages” to configure GitHub Pages

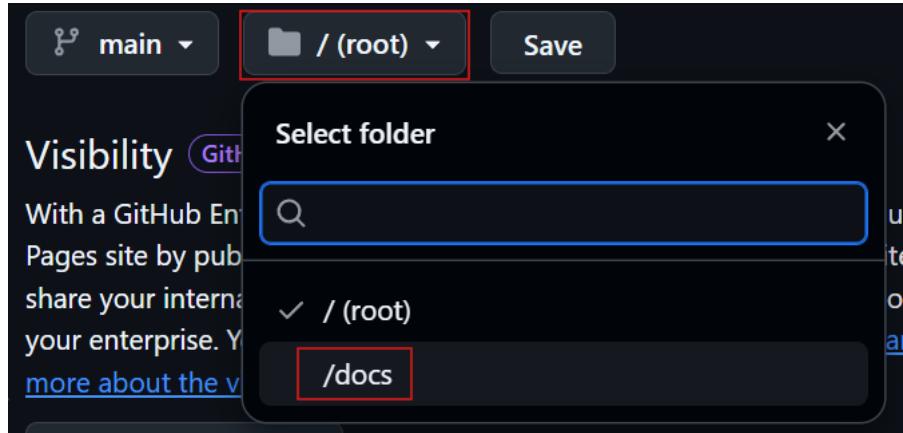
Speaker Notes - Now to put everything together. We need to create a new repository on GitHub which is where our project files will be held. - Then, we connect our local project to GitHub using `git remote add origin` followed by the repo’s HTTPS link (found on the GitHub website). This tells Git where to send the files when we push them with `git push -u origin main`. *Instructor Notes* - For this to work, the repo created must be set to Public

Practical exercise 11

- Change “none” to “main”



- Change “/root” to “/docs”
- Click Save



- After 1-2 minutes, you will see that your site will be **live!**

Speaker Notes - The last step to make the site public is to configure the GitHub repository settings under “Pages.” - Follow the steps to switch to the branch we are working on and switch the folder where Quarto rendered the website files. *Instructor Notes* - Reminder that the citations will not be complete because configuring to include the bibliography was not done. That’s an additional step.

Recap

About Quarto

- Quarto is a powerful tool that combines text, code, and media to make documents which are transparent, reproducible, and accessible.
- In a **.qmd** file:
 - The **YAML header** controls the document settings
 - **Code chunks** run code and show results
 - **Markdown text** creates the main content
- You can use simple formatting and keyboard shortcuts for styling text.
 - ****bold**** → **bold**
 - ***italic*** → *italic*

Speaker Note - Here are some of the key things covered in the lesson that give a general overview of Quarto and what it contains. - Quarto is more than just a tool to write documents, it’s designed to bring together text, code, and media in one .qmd file that ensures transparency and reproducibility. - Quarto supports simple formatting, like bold and italics, which makes writing cleaner and more efficient without the need for complex styling tools.

Instructor Note - Aim: to give a recap of important elements of Quarto covered in the lesson.

Recap

Features and Sharing

- Code chunks can be customized in the YAML header (for example, `code-fold: true` makes code collapsible).
- You can add **links, images, media, and layout features** (like columns) to improve presentation and usability.
- **Zotero** helps manage references, allowing you to cite while you write and automatically builds a bibliography.
- **Publishing** makes your work accessible and reproducible.
 - Options for publishing include **Quarto Pub**, **GitHub Pages**, or your own **website**.

For a summary of what we learned and applied using Quarto, click [here](#) to download a cheatsheet.

Speaker Note - Here are some of the features for customization and improved usability that were discussed and applied in the lesson to create the Quarto document. We then shared it using GitHub Pages. - Zotero integration streamlines citation management and bibliography building, which is essential for academic and research writing. - Quarto enables sharing work through platforms like Quarto Pub or GitHub Pages, which makes research accessible and reproducible. - For quick reference, a downloadable cheatsheet is provided with this lesson. It offers beginner-friendly instructions summarizing the key terms, features, and functions of Quarto. *Instructor Note* - Aim: to give a recap of things you can do to edit the Quarto document and then make it shareable. - Instruct students to download a cheatsheet to have a summary of information and features that were covered in this session.

From SvG: Would this differ from the take home messages? Yes- I made the recap like an overall review that's instructor-led and the take home message as an opportunity for students to reflect on the lesson.

Relevance and implications

Let's discuss how useful **Quarto** can be for **open research**. Consider the following questions:

- How can using Quarto improve transparency and reproducibility in your work?
- In what ways does Quarto make sharing code and data with collaborators easier?

- Do you see advantages in using Quarto compared to traditional tools like Word or PowerPoint?
- What challenges might you face in adopting Quarto in your current

Speaker Note - Let's discuss how the new skills could be applied in practise with specific examples. Use the questions on the slide to guide the discussion. *Instructor Note* - Aim: To work out the relevance of the topic to your students, examine downfalls, and practical obstacles in an interactive setting.

What is the take-home message?

Let's wrap up by identifying the **key takeaway** from today's session together.

Ask yourself: "*If you had to explain to a colleague in one sentence why learning Quarto matters for research and teaching, what would you say?*"

Speaker Notes - Encourage students to share one sentence take-home messages. You can collect them verbally or in a shared doc/board. *Instructor Notes* - Aim: End lesson on clear take-home message that are interactively compiled by students. - Reinforce that Quarto is not just a tool, but a mindset shift toward openness and collaboration.

Assignment

After learning about Quarto in today's lesson, use this homework assignment to:

- Practice your Quarto skills
- Explore what more can be added to your documents
- Build confidence in editing YAML header, adding code chunks, and customizing the layout

For the assignment, click [here](#) to download the homework sheet.

Speaker Notes - - Aim: Practice additional Quarto features. The homework activity builds on the document created during this lesson to add more to the content and layout. - The homework contains activities that take what we practiced today to the next level. Each item builds on the knowledge and exercises that were covered in this session.

Instructor Note - Aim: to reinforce what was learned in this session and to have an opportunity to apply the knowledge in a more challenging way. - Use this homework sheet as it is or use it as an example to create your own based on the needs of the class or the specific requirements for future lessons. - Examine whether/how it will be assessed - Mention scoring

rubrics, if applicable - Design a peer-review system for assignments to place students in role of reviewer and author

To conclude: Survey time!

Let's end this session with a short survey on topics covered in this lesson.

Speaker Notes - To end this lesson, let's use this survey to see where we stand in our knowledge of Quarto and what we can focus on more moving forward to become more proficient in literate programming with it. *Instructor Notes* - Aim: This post-submodule survey serves to examine students' current knowledge about the submodule's topic. - Use free survey software such as or other survey software (particify, formR) to establish the following questions (these are examples that can be adapted to suit your needs):

What is your level of familiarity with Quarto before this lesson?

- a) I've never used it
 - b) I've heard of it but never tried it
 - c) I've experimented a little
 - d) I use it occasionally
 - e) I use it frequently
-

Which of the following best describes Quarto?

- a) A programming language
 - b) A publishing system for reproducible documents
 - c) A code editor
 - d) A type of version control system
-

Which components are combined in the authoring process of Quarto documents?

- a) Only YAML
- b) Only Markdown
- c) Neither

- d) YAML and Markdown together
-

Which of the following modes can be used to create a Quarto document?

- a) Source mode
 - b) Visual mode
 - c) Both
 - d) Neither
-

On a scale of 1 to 5, how comfortable are you switching between Source and Visual modes?

Which of these are examples of Markdown text formatting? (Select all that apply)

- a) Bold text
 - b) Italic text
 - c) Adding code chunks
 - d) Adding hyperlinks
-

Why should you use text formatting in Quarto documents?

- a) For aesthetics only
 - b) To improve clarity and emphasis
 - c) To make the file larger
 - d) To change the coding language
-

When would you consider using multi-column layouts?

- a) To shorten the document
- b) To make information more visually organized
- c) To add footnotes only
- d) To increase file size

What is the purpose of code chunks in Quarto documents?

- a) To include executable code and results within the document
 - b) To store references
 - c) To format text
 - d) To hide metadata
-

Which YAML options relate to how code chunks are displayed?

- a) echo: true
 - b) code-fold: false
 - c) title: "My Document"
 - d) format: html
-

Why is including code in research documents important?

- a) To make the document more difficult to read
 - b) To ensure only the author can reproduce results
 - c) To replace data collection
 - d) To make research reproducible and transparent
-

**Which of the following are common publishing options for Quarto projects?
(Select all that apply)**

- a) GitHub Pages
 - b) Quarto Pub
 - c) Social media platforms
 - d) Personal websites
-

What is the benefit of publishing your Quarto documents?

- a) They make your computer run faster
 - b) They allow offline editing only
 - c) They become accessible to others for sharing and collaboration
 - d) They automatically write the code for you
-

Discussion of survey results

Speaker Notes - To end off, we can use the answers from this survey to see how much we know about Quarto after this lesson and where we should focus and explore more moving forward if we are to become proficient in using it. *Instructor Notes* - Aim: Briefly examine the answers given to each question interactively with the group. - Compare and highlight specific differences in answers between pre- and post-survey answers

References

- Li, J., Yuan, J., Miao, Z., Song, Z., Yang, Y., Tian, W., & Guo, Y. (2016). *Effect of dietary nutrient density on small intestinal phosphate transport and bone mineralization of broilers during the growing period.* PLOS ONE, 11(4), e0153859. <https://doi.org/10.1371/journal.pone.0153859>
- Pauwels, J., Coopman, F., Cools, A., Michiels, J., Fremaut, D., De Smet, S., & Janssens, G. P. J. (2015). *Selection for growth performance in broiler chickens associates with less diet flexibility.* PLOS ONE, 10(6), e0127819. <https://doi.org/10.1371/journal.pone.0127819>
- Forehand, M. (2010). Bloom's taxonomy. *Emerging perspectives on learning, teaching, and technology*, 41(4), 47-56.

From SvG: Add references here Beth: should I cite the Quarto Website?

Thanks!

See you next class :)

Pedagogical add-on tools for instructors

- Relevant practical exercises follow each section to encourage students to learn-by-doing
- Downloadable cheatsheet as a PDF for a compact summary of all the topics covered in the session
- Downloadable homework assignment sheet as a PDF to learn additional Quarto features, designed with slightly less guidance so that students are encouraged to apply their knowledge and skills more independently.
- [Link to advanced Quarto activities](#) for faster learners who want to gain deeper knowledge on Quarto and its possibilities

From SvG: Still to do right? Beth: yes

Additional literature for instructors

- References for content
 - <https://quarto.org/docs/guide/>
- References for pedagogical add-on tools
 - <https://quarto.org/docs/presentations/>
 - Bloom's Taxonomy
- Other resources
 - [Video on crafting presentations with R and Quarto](#)

From SvG: Still to do right? Beth: this video was helpful but I did not incorporate everything from it
