

Introduction to R, RStudio, and Quarto

Emile Latour, Nicky Wakim, Meike Niederhausen

January 5, 2026

Learning objectives (today)

By the end of class, you should be able to:

1. Open and navigate RStudio
2. Create, edit, save, and render a Quarto (.qmd) document
3. Use basic Markdown to format text in a Quarto document
4. Create and run simple R code chunks and render the output to HTML

Introduction to R



Artwork by @allison_horst

What is R?

- A programming language
- Focus on statistical modeling and data analysis
 - import data, manipulate data, run statistics, make plots
- Useful for data science
- Great visualizations
- Also useful for most anything else you'd want to tell a computer to do
- Interfaces with other languages i.e. python, C++, bash



For the history and details: [Wikipedia](#)

- an interpreted language (run it through a command line)
- procedural programming with functions
- Why "R"?? Scheme inspired S (invented at Bell Labs in 1976) which inspired R since 1st letters of original authors (**free and open source!** in 2000)

What is RStudio?

R is a programming language

RStudio is an integrated development environment (IDE)
= an interface to use R (with perks!)

R: Engine



RStudio: Dashboard



Open RStudio on your computer (not R!)

1.1.2 Using R via RStudio

Recall our car analogy from earlier. Much as we don't drive a car by interacting directly with the engine but rather by interacting with elements on the car's dashboard, we won't be using R directly but rather we will use RStudio's interface. After you install R and RStudio on your computer, you'll have two new *programs* (also called *applications*) you can open. We'll always work in RStudio and not in the R application. Figure 1.2 shows what icon you should be clicking on your computer.

R: Do not open this



RStudio: Open this



FIGURE 1.2: Icons of R versus RStudio on your computer.

RStudio anatomy

The screenshot shows the RStudio interface with various components highlighted by orange boxes and arrows:

- Script file:** A panel on the left where code is written. It includes instructions for running code, assignment operators (<-), and comments.
- Console:** A panel at the bottom where code is run and output is displayed. It shows the R prompt (>), command history, and output results.
- Environment:** A panel on the right showing the global environment with objects like mx and x.
- History:** A panel on the right showing a history of commands run in the console.
- Packages:** A panel on the right showing the system library with installed packages like abind, acepack, ade4, agricolae, and AlgDesign.
- Plots:** A panel on the right showing figure output.

Annotations provide additional details:

- Script file:** Write code here. To run code put your cursor on the line and click the run button. Edit to correct errors.
 - ⇒ record of commands that worked
 - Save scripts with the .R extension
 - ⇒ syntax will be highlighted
 - ⇒ good practice
 - <- is the assignment operator
 - ⇒ puts what is on the right in to the object on the left
 - ⇒ Assign results if you want to use them again
- Console:** When you click run, code is sent to the console and executed.
 - > is the prompt
 - ⇒ do not type it
 - ⇒ appears when R is ready for next command
 - Command output goes here by default
 - ⇒ output is in a different colour
 - ⇒ [1] indicates 3.4 is the first element of the output
 - ⇒ many commands will not have output, the prompt just reappears
- Environment:** Name objects by assignment to use them again. All the objects you created in your session. Saving the environment saves all the objects, but not the code with a .RData extension.
- History:** A history of every command you sent to the console, mistakes included. File can be saved but usually you just need the script.
- Packages:** Many functions come with R. A huge amount of extra functionality is available in packages. Packages can be installed by clicking the Install button.
- Plots:** Access to manual pages for all installed packages. Figure output appears here.

Emma Rand

Read more about RStudio's layout in Section 3.4 of “**Getting Used to R, RStudio, and R Markdown**” (Ismay and Kennedy 2016)

Saving your work with Quarto

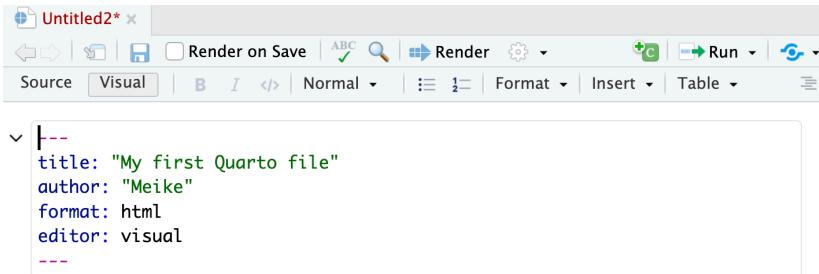
or, creating reproducible reports



Artwork by @allison_horst

Example: creating an html file

.qmd file



```
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

.html output

My first Quarto file

AUTHOR

Meike

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

[1] 2

You can add options to executable code like this

```
[1] 4
```

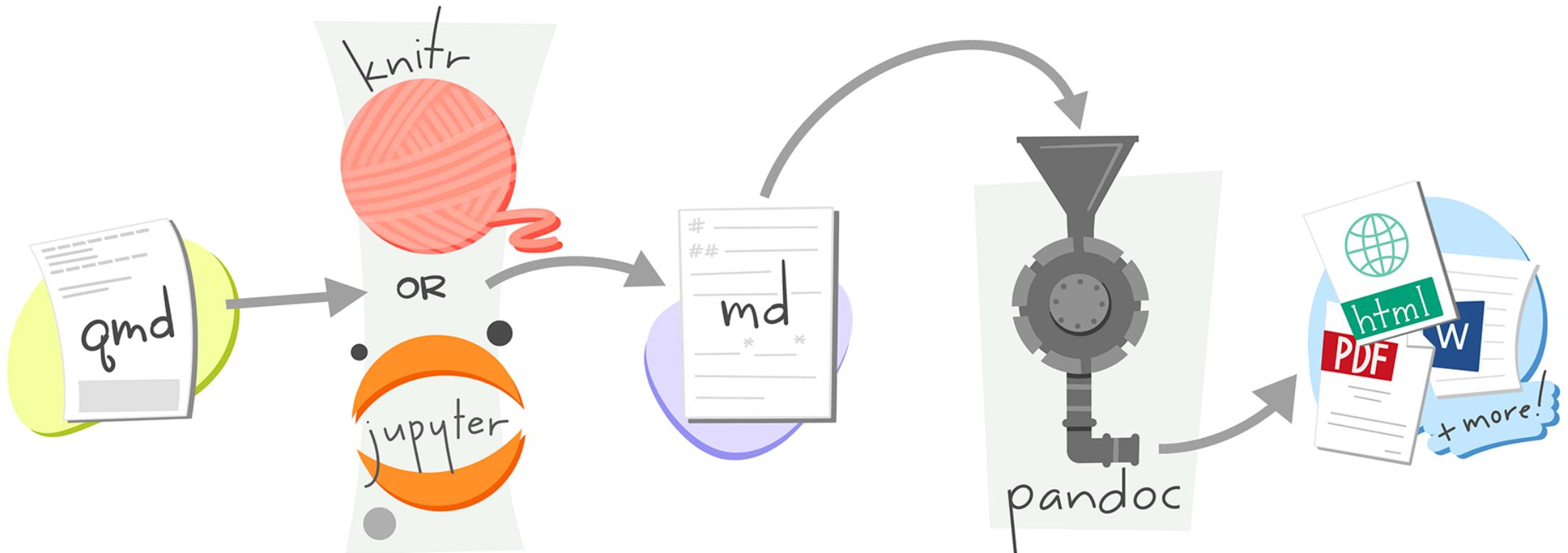
The `echo: false` option disables the printing of code (only output is displayed).

Quarto = .qmd file = Code + text

A `.qmd` file is where you **write** your work.

Rendering turns it into something you can **read and submit**.

If you edit the `.qmd`, re-render to update the `.html`.



Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

Basic Quarto example



Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

Before we get further in .qmd files

- If you would like to code along, make sure to have RStudio open.

Steps for making a Quarto file

1. Create a Quarto file ([.qmd](#))
2. Edit a Quarto file ([.qmd](#))
3. Save the Quarto file ([.qmd](#))
4. Create html file

1. Create a Quarto file (.qmd)

Two options:

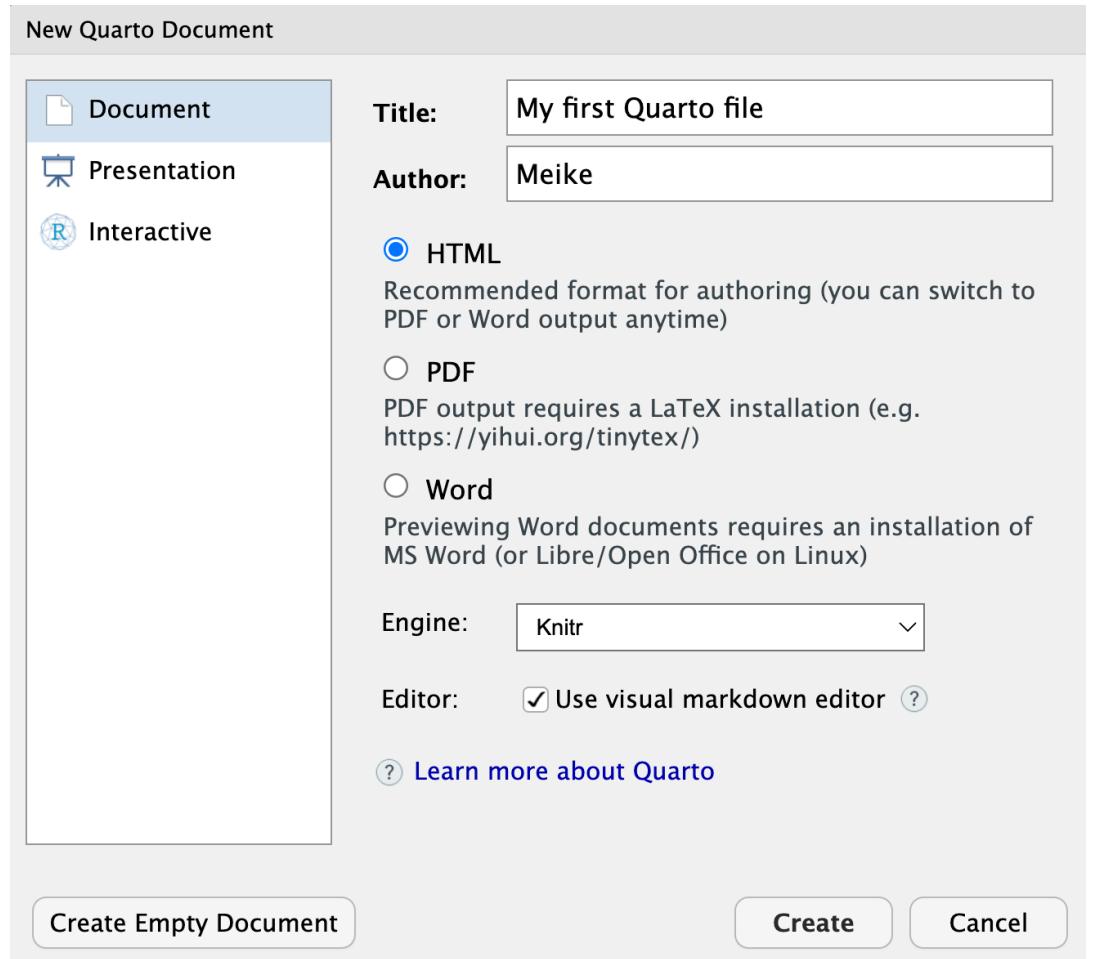
1. click on File → New File → Quarto Document... → OK,

2. or in upper left corner of RStudio click on  →

 Quarto Document...

Pop-up window selections:

- Enter a title and your name
- Select **HTML** output format (default)
- Engine: select **Knitr**
- Editor: Select **Use visual markdown editor**
- Click **Create**



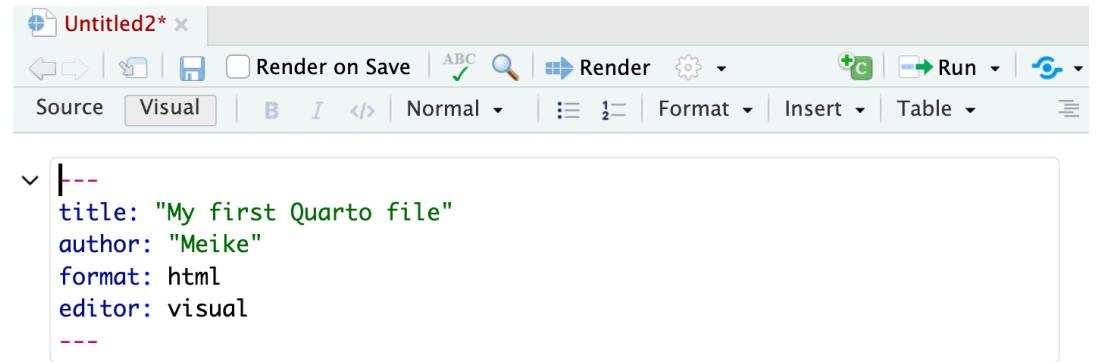
The top block (YAML) - for later

That block controls document settings.

We will come back to it later. For now: **do not change it.**

2. Edit a Quarto file (.qmd)

- After clicking on **Create**, you should then see the following in your editor window:
- You can try editing the text or changing the code!
 - Make sure you are only editing at the “Quarto” header and below



The screenshot shows the RStudio interface with a Quarto file titled "Untitled2.qmd". The header section contains the following configuration:

```
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

3. Save the Quarto file (. qmd)

- **Save the file** by
 - selecting **File** → **Save**,
 - or clicking on  (towards the left above the scripting window),
 - or keyboard shortcut
 - PC: *Ctrl + s*
 - Mac: *Command + s*
- You will need to specify
 - a **filename** to save the file as
 - ALWAYS use **.qmd** as the filename extension for Quarto files
 - the **folder** to save the file in
 - For now, save it to your desktop, or wherever you know you will find it again.

4. Create html file

We create the html file by **rendering** the .qmd file.

Two options:

1. click on the Render icon  at the top of the editor window,

2. or use keyboard shortcuts

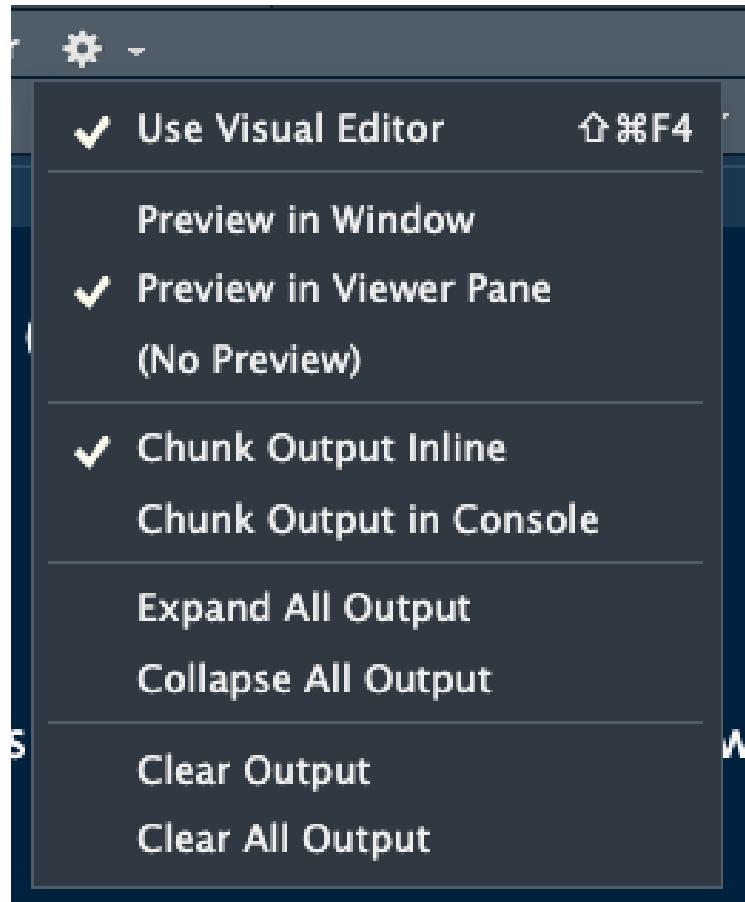
- Mac: *Command+Shift+K*
- PC: *Ctrl+Shift+K*
- A new window will open with the html output.
- You will now see both .qmd and .html files in the folder where you saved the .qmd file.

 **Note**

- The template .qmd file that RStudio creates will render to an html file by default.
- The output format can be changed to create a Word doc, pdf, slides, etc.

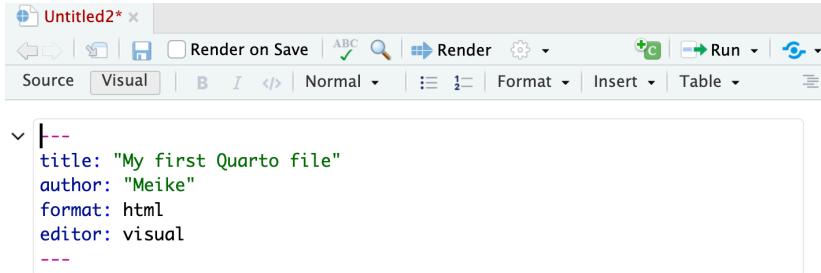
Tip: changing the render view

- You can change where your `.html` file pops up
- I have it set to open in the “Viewer Pane” in the bottom right



.qmd vs. its .html output

.qmd file



```
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

.html output

My first Quarto file

AUTHOR

Meike

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

[1] 2

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

File naming for assignments

Use a consistent naming convention so we can grade efficiently:

`Lastname_FirstInitial_HW00.qmd`

Submit **both**: - .qmd - .html

3 types of Quarto content

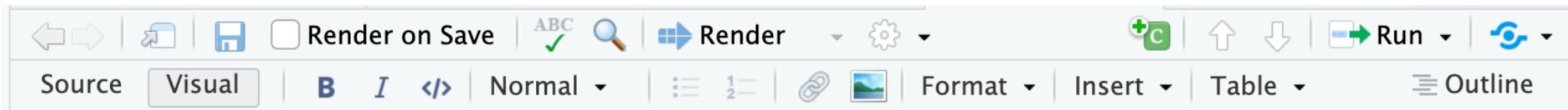
1. *Text, lists, images, tables, links*
2. Code chunks
3. YAML metadata



Illustration by Alison Hill and Allison Horst, for RStudio.

Formatting text

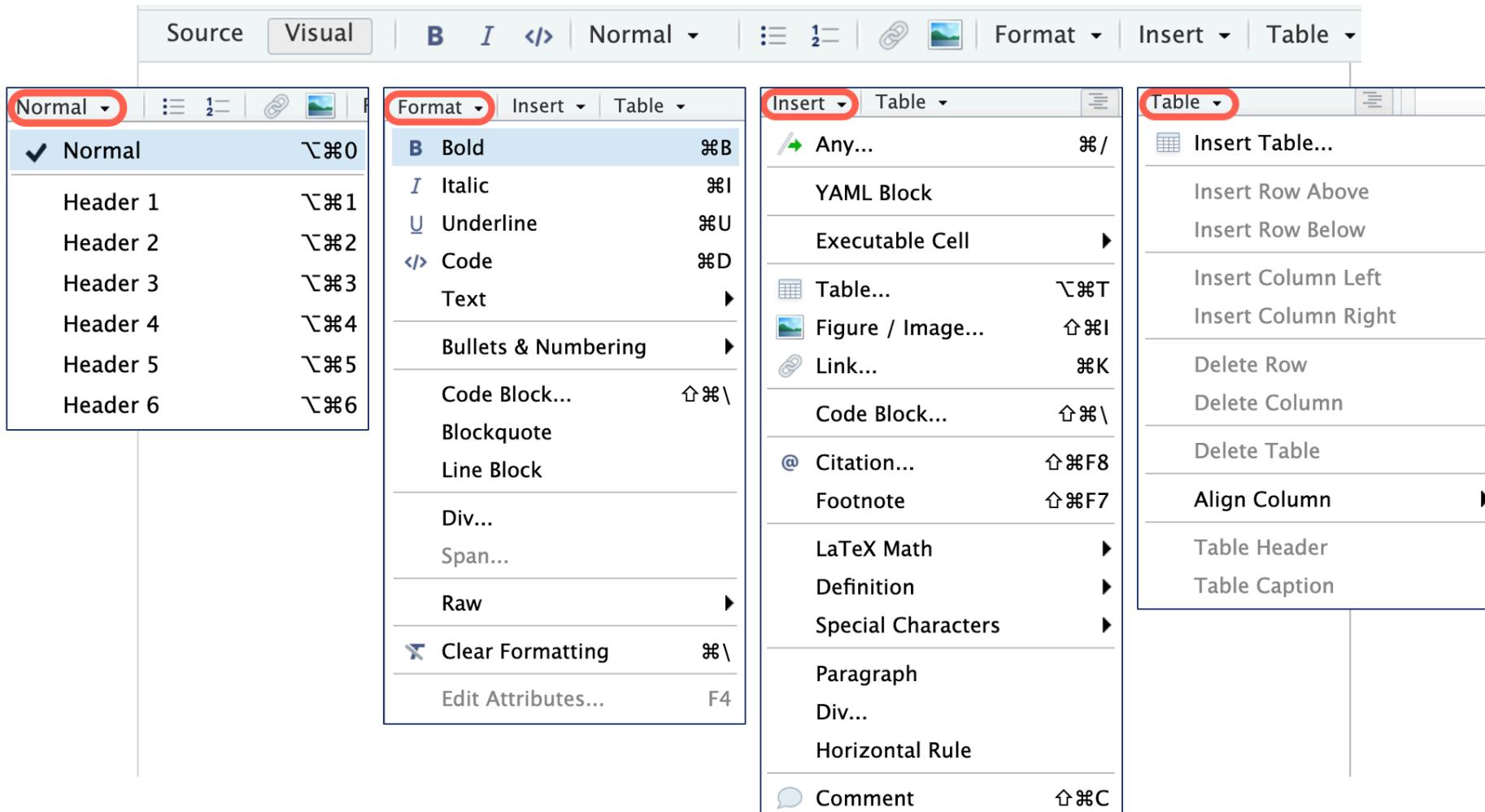
- **bold**, *italics*, super^{scripts} & sub_{scripts}, ~~strike~~through, **verbatim**, etc.
- Text is formatted through a markup language called **Markdown** ([Wikipedia](#))
 - Other markup languages include html (webapges) and LaTeX (math)
 - All text formatting is specified via code
 - "Markdown is a plain text format that is designed to be easy to write, and, even more importantly, easy to read" ¹
- Newer versions of RStudio include a **Visual editor** as well that makes formatting text similar to using a word processor.



1. From Quarto's Markdown Basics webpage, <https://quarto.org/docs/authoring/markdown-basics.html>

Formatting text: Visual editor

- Using the **Visual editor** is similar to using a wordprocessor, such as Word
- Keyboard shortcuts usually work as well (*shown for Mac below*)



Practice

1. Part 1

1. Using the visual editor, practice formatting text in your qmd file, such as making text **bold**, *italicized*, and in `code` format.
2. Add 1st, 2nd, and 3rd level headers
3. Add a list with a
 - sub-list (bullet and/or numbered)
4. Add a table
5. Add whatever else you are interested in!

2. Part 2

1. Switch back to the [Source](#) editor and examine the markdown code that was used for the formatting.

Questions:

1. What went smoothly?
2. What hurdles did you encounter?

How this connects to Homework 0

Everything you practiced today is exactly what you need for HW 0:

- Editing text with Markdown
- Creating and running R code chunks
- Saving a `.qmd` file
- Rendering to an `.html` file

For Homework 0, you will:

- Start from a provided `.qmd` template
- Make small edits
- Render successfully
- Submit **both** the `.qmd` and `.html`

Formatting text: Markdown

Markdown:

This text is in italics, but _so is this text_.

****Bold**** also has __2 options__

~~Should this be deleted?~~

Need^{super}^ or_{sub}~ scripts?

`Code is often formatted as verbatim`

>This is a block quote.

Output:

This text is in italics, but so is this text.

Bold also has **2 options**

~~Should this be deleted?~~

Need^{super} or_{sub} scripts?

Code is often formatted as verbatim

This is a block quote.

Headers

- Organize your documents using headers to create sections and subsections
- Use `#` at the beginning of the line to create headers

Text in editor:

```
# Header 1  
## Header 2  
### Header 3  
#### Header 4  
##### Header 5  
##### Header 6
```

Output:

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

 **Important**

Make sure there is no space before the `#`, and there IS a space after the `#` in order for the header to work properly.

RStudio tip

You can easily navigate through your `.qmd` file if you use headers to outline your text

The screenshot shows the RStudio interface with a .qmd file open. The top menu bar includes 'Render on Save' (unchecked), 'Render' (button), and 'Outline' (button). The toolbar below has icons for back, forward, search, and various document operations. The main workspace displays the following content:

```
---
```

```
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
---
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

3 types of Quarto content

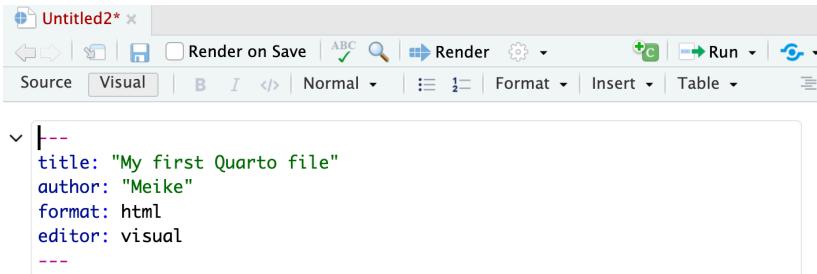
1. Text, lists, images, tables, links
2. *Code chunks*
3. YAML metadata



Illustration by Alison Hill and Allison Horst, for RStudio.

Code chunks

.qmd file



```
title: "My first Quarto file"
author: "Meike"
format: html
editor: visual
```

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}
1 + 1
```

You can add options to executable code like this

```
{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

.html output

My first Quarto file

AUTHOR

Meike

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

Create a code chunk

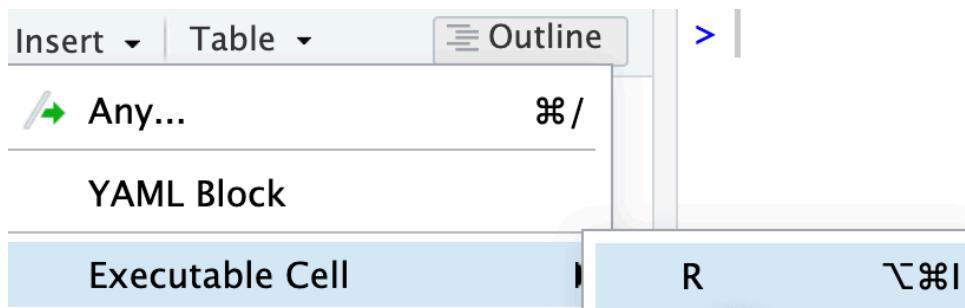
3 options to create a code chunk

1. Click on  at top right of the editor window, or

2. **Keyboard shortcut**

Mac	<i>Command + Option + I</i>
PC	<i>Ctrl + Alt + I</i>

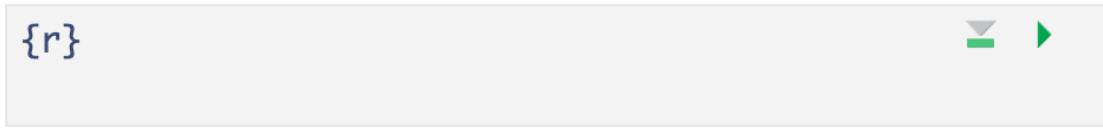
3. **Visual editor:** Select **Insert** -> **Executable Cell** -> **R**



What does a code chunk look like?

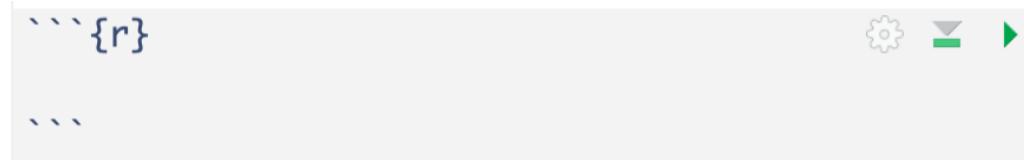
An empty code chunk looks like this:

Visual editor



A screenshot of a visual editor interface. It shows a single line of code: '{r}'. To the right of the code are three icons: a downward arrow, a green checkmark, and a green right-pointing arrow.

Source editor



A screenshot of a source editor interface. It shows the code chunk starting with '```{r}' followed by three blank lines. To the right of the code are three icons: a gear, a downward arrow, and a green right-pointing arrow.



Important

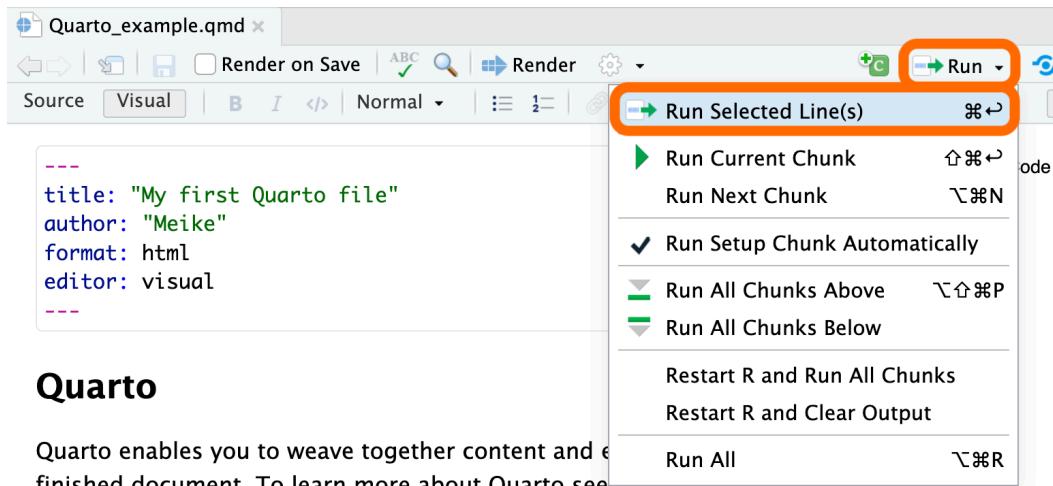
Note that a code chunks start with ```{r} and ends with ```. Make sure there is no space before ```.

Enter and run code (1/n)

- Type R code inside code chunks
- Select code you want to run, by
 - placing the cursor in the line of code you want to run,
 - or highlighting the code you want to run
- Run selected code by
 - clicking on the  button in the top right corner of the scripting window and choosing **Run Selected Line(s)**,
 - or typing one of the following key combinations:

Mac **ctrl + return**
PC **command + return**

- Where does the output appear?



Quarto

Quarto enables you to weave together content and code into a single, finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}  
1 + 1
```

Enter and run code (2/n)

- Run all code in a chunk by
 - by clicking the play button in the top right corner of the chunk
- The code output appears below the code chunk

```
{r}
1 + 1
```

```
[1] 2
```

Runs all code in chunk

Runs all code in previous chunks

Note

- The output should also appear in the Console.
- Settings can be changed so that the output appears only in the Console and not below the code chunk:
 - Select (to right of Render button) and then *Chunk Output in Console*.

Useful keyboard shortcuts

[Full list of keyboard shortcuts](#)

action	mac	windows/linux
Run code in qmd (or script)	cmd + enter	ctrl + enter
<-	option + -	alt + -
interrupt currently running command	esc	esc
in console, retrieve previously run code	up/down	up/down
keyboard shortcut help	option + shift + k	alt + shift + k

Practice

Try typing code below in your qmd (with shortcut) and evaluating it:

```
1 y <- 5  
2 y
```

3 types of Quarto content

1. Text, lists, images, tables, links
2. Code chunks
3. *YAML metadata*



Illustration by Alison Hill and Allison Horst, for RStudio.

YAML metadata

The block at the top controls document settings. We'll come back to it later. For now, don't touch it.