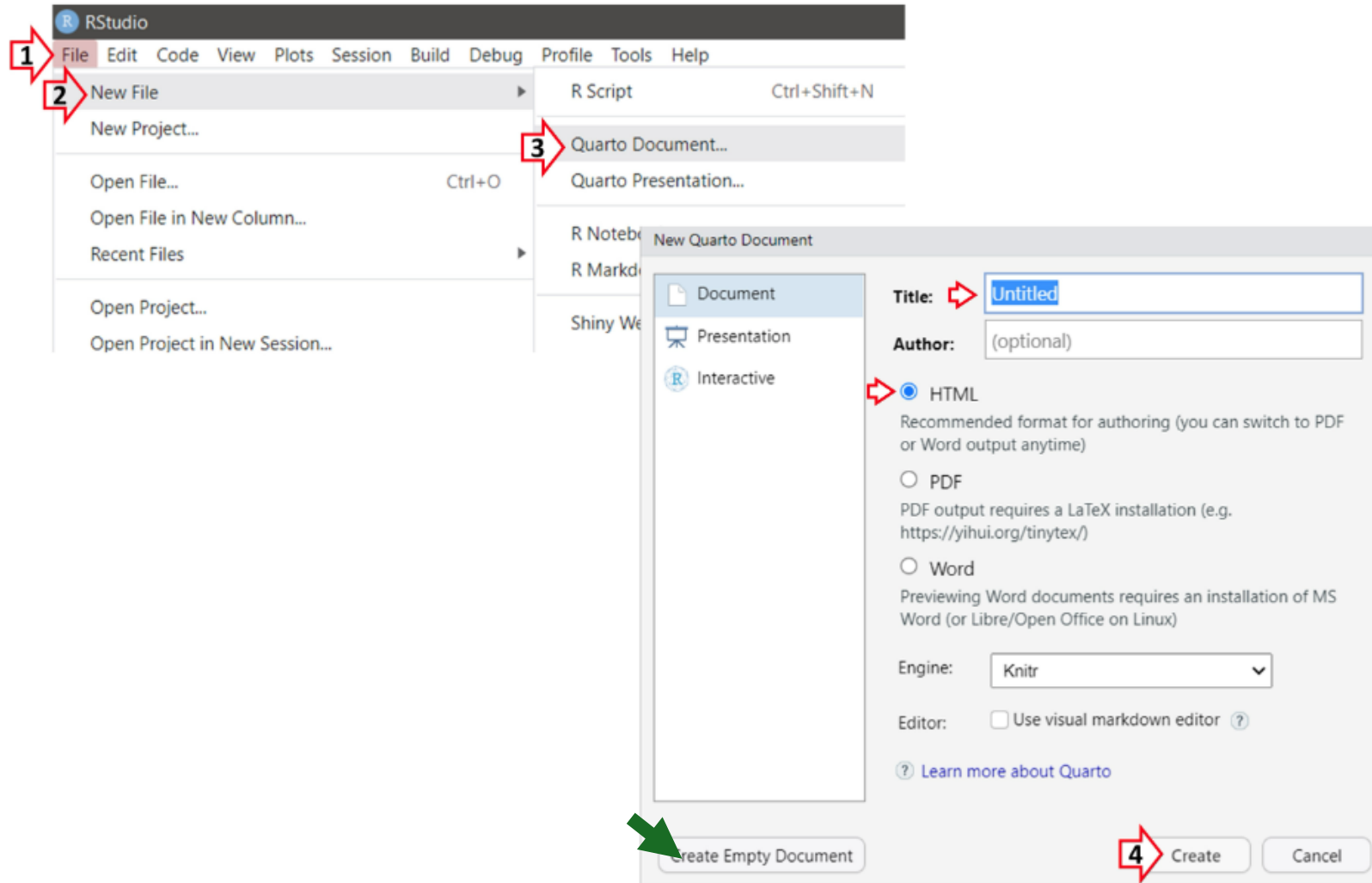


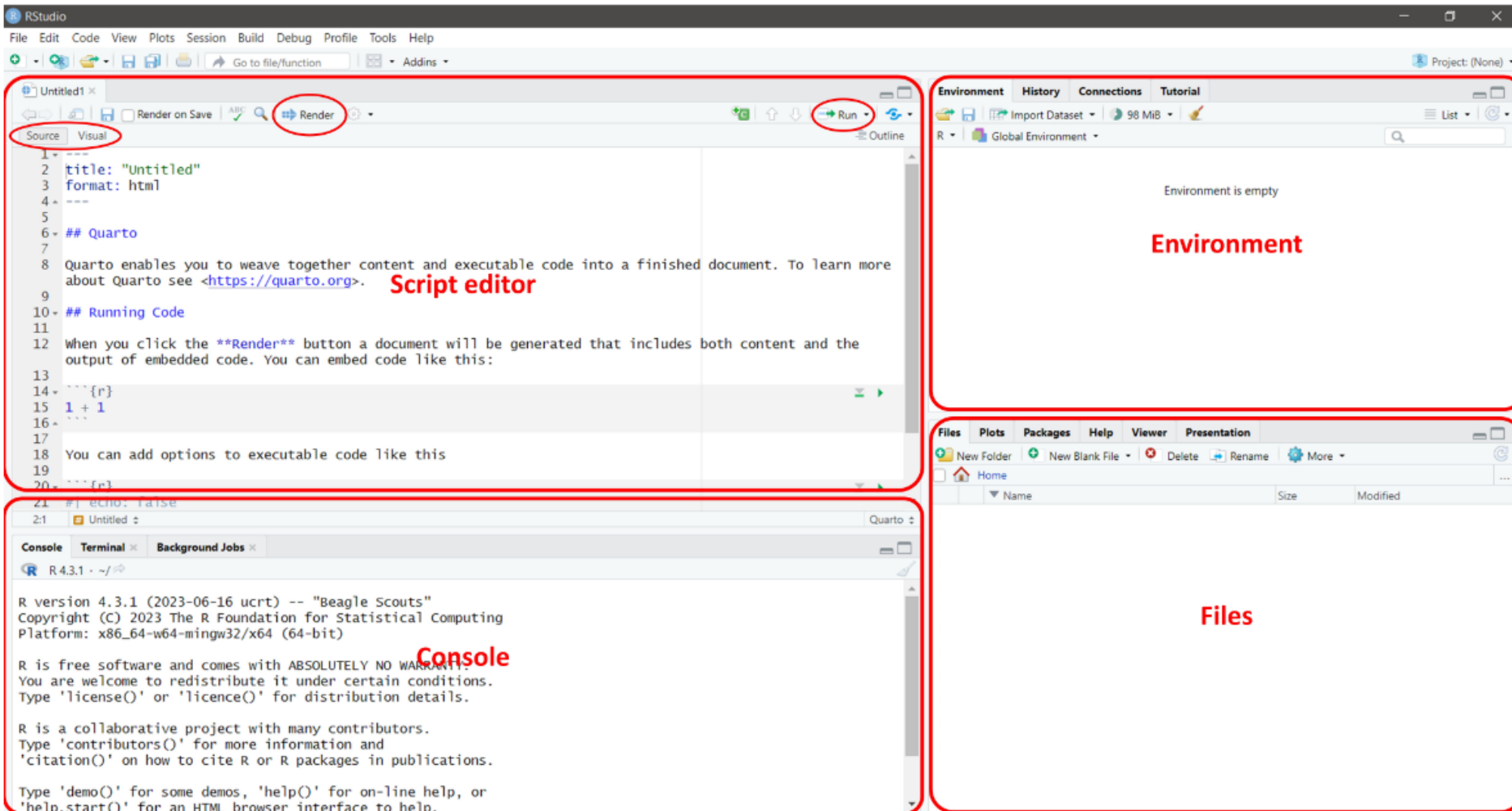
# Getting started with Quarto document

# 1 - New Quarto Document:



# 1 - New Quarto Document:

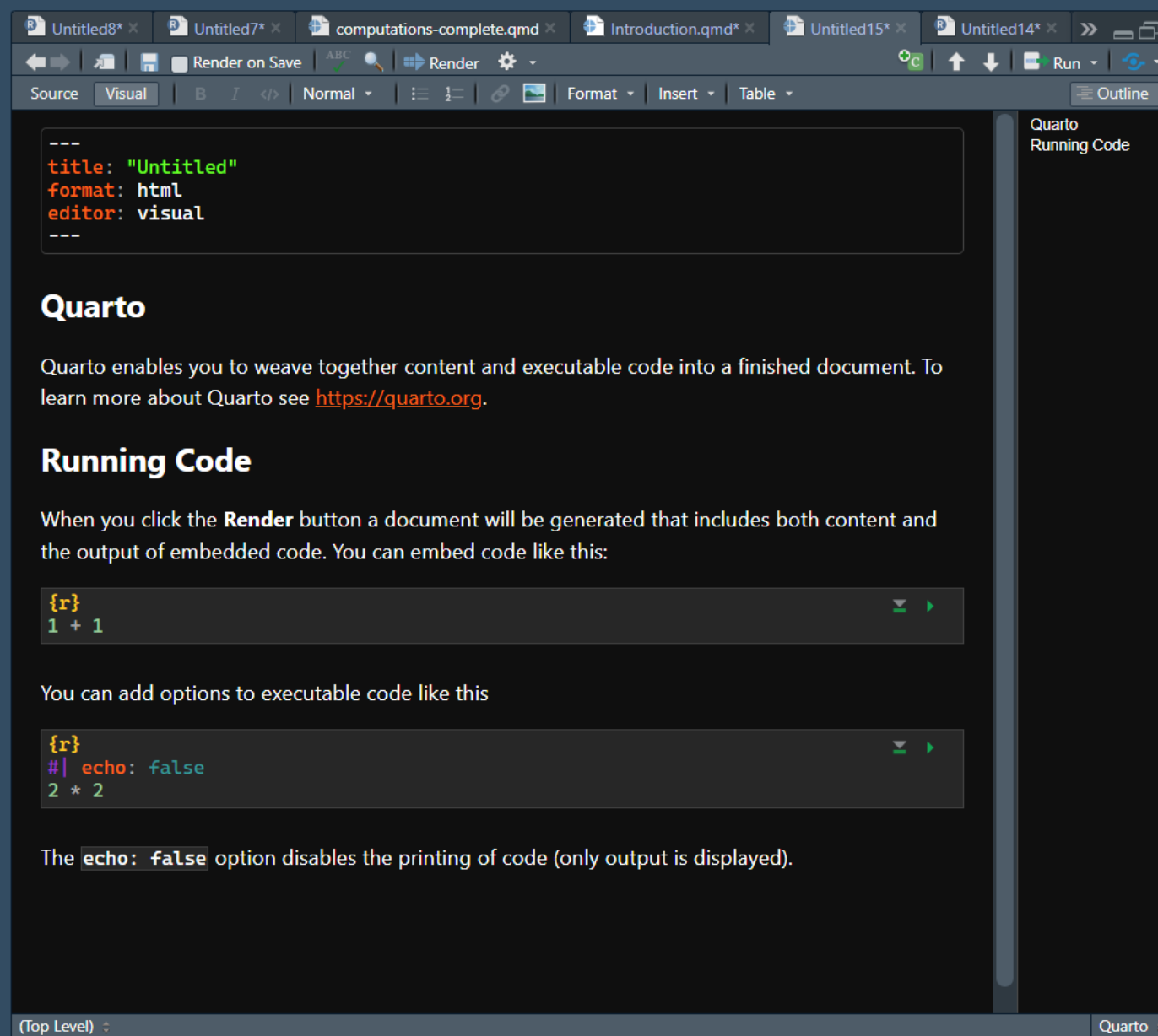
Create a file with temporary content to start



... save the document

... a start for your future project

# < New document



The screenshot shows the Quarto editor interface. At the top, there are several tabs for documents: 'Untitled8\*', 'Untitled7\*', 'computations-complete.qmd', 'Introduction.qmd', 'Untitled15\*', and 'Untitled14\*'. Below the tabs is a toolbar with icons for navigation and actions like 'Render on Save', 'Render', and 'Run'. The main editor area is divided into two sections: 'Source' and 'Visual'. The 'Source' section shows a YAML header:

```
---  
title: "Untitled"  
format: html  
editor: visual  
---
```

The 'Visual' section displays the rendered content. It starts with the title 'Quarto' in a large, bold font. Below the title is a paragraph: 'Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.' This is followed by another section header 'Running Code'. Below this header is a paragraph: '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:'. There are two code blocks shown. The first block contains the code `{r}` followed by `1 + 1`. The second block contains the code `{r}` followed by `#| echo: false` and `2 * 2`. On the right side of the editor, there is a sidebar titled 'Quarto Running Code'. At the bottom of the editor, there is a status bar showing '(Top Level)' on the left and 'Quarto' on the right.

# < Visual

---  
title: "Introduction to Quarto"  
author: "Ntsoa"  
format:  
 html: default  
 pdf: default  
 editor: visual  
---

### Working directory

```
{r}  
wd ← "C:/Users/rntso/OneDrive/Bureau/Quarto intro"  
setwd(wd)
```

### 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
```

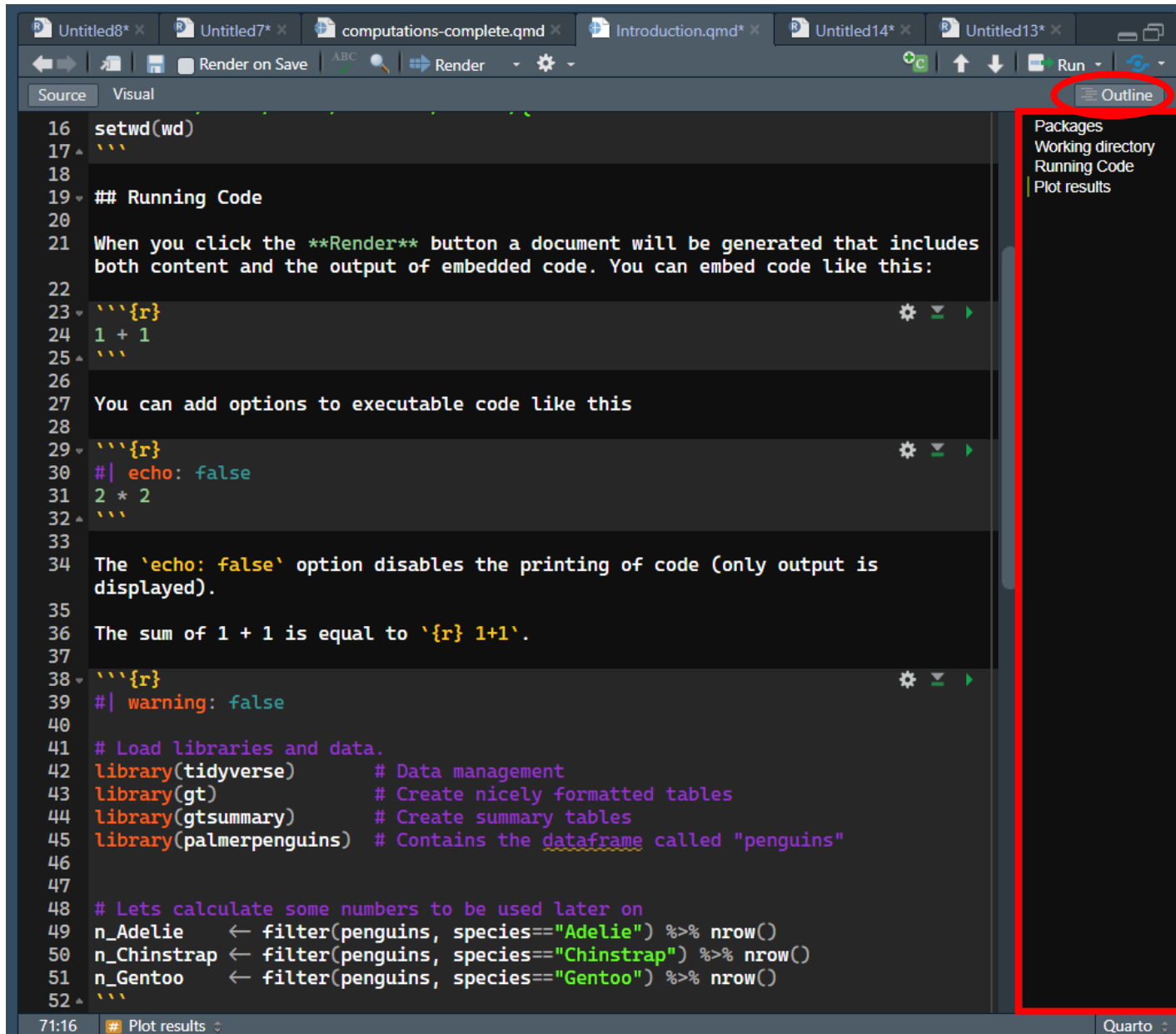
The `echo: false` option disables the printing of code (only output is displayed).

The sum of 1 + 1 is equal to `{r} 1+1`.

# < Source

```
1 ---  
2 title: "Introduction to Quarto"  
3 author: "Ntsoa"  
4 format:  
5   html: default  
6   pdf: default  
7   editor: visual  
8 ---  
9  
10 ## Working directory  
11  
12 ```{r}  
13 wd ← "C:/Users/rntso/OneDrive/Bureau/Quarto intro"  
14 setwd(wd)  
15 ```  
16  
17 ## Running Code  
18  
19 When you click the Render button a document will be generated that includes both  
20 content and the output of embedded code. You can embed code like this:  
21  
22 ```{r}  
23 1 + 1  
24 ```  
25  
26 You can add options to executable code like this  
27  
28 ```{r}  
29 #| echo: false  
30 2 * 2  
31 ```  
32  
33 The echo: false option disables the printing of code (only output is displayed).  
34  
35 The sum of 1 + 1 is equal to {r} 1+1.  
36  
37 ```{r}  
38 #| warning: false  
39 ```
```

# < Outline



The image shows a screenshot of the Quarto editor interface. The main window displays a code document with the following content:

```
16 setwd(wd)
17 ```
18
19 ## Running Code
20
21 When you click the Render button a document will be generated that includes
22 both content and the output of embedded code. You can embed code like this:
23 ```{r}
24 1 + 1
25 ```
26
27 You can add options to executable code like this
28
29 ```{r}
30 #| echo: false
31 2 * 2
32 ```
33
34 The `echo: false` option disables the printing of code (only output is
35 displayed).
36
37 The sum of 1 + 1 is equal to `{r} 1+1`.
38 ```{r}
39 #| warning: false
40
41 # Load libraries and data.
42 library(tidyverse) # Data management
43 library(gt) # Create nicely formatted tables
44 library(gtsummary) # Create summary tables
45 library(palmerpenguins) # Contains the dataframe called "penguins"
46
47
48 # Lets calculate some numbers to be used later on
49 n_Adelie <- filter(penguins, species=="Adelie") %>% nrow()
50 n_Chinstrap <- filter(penguins, species=="Chinstrap") %>% nrow()
51 n_Gentoo <- filter(penguins, species=="Gentoo") %>% nrow()
52 ```
```

The Outline panel on the right side of the editor is highlighted with a red border and contains the following items:

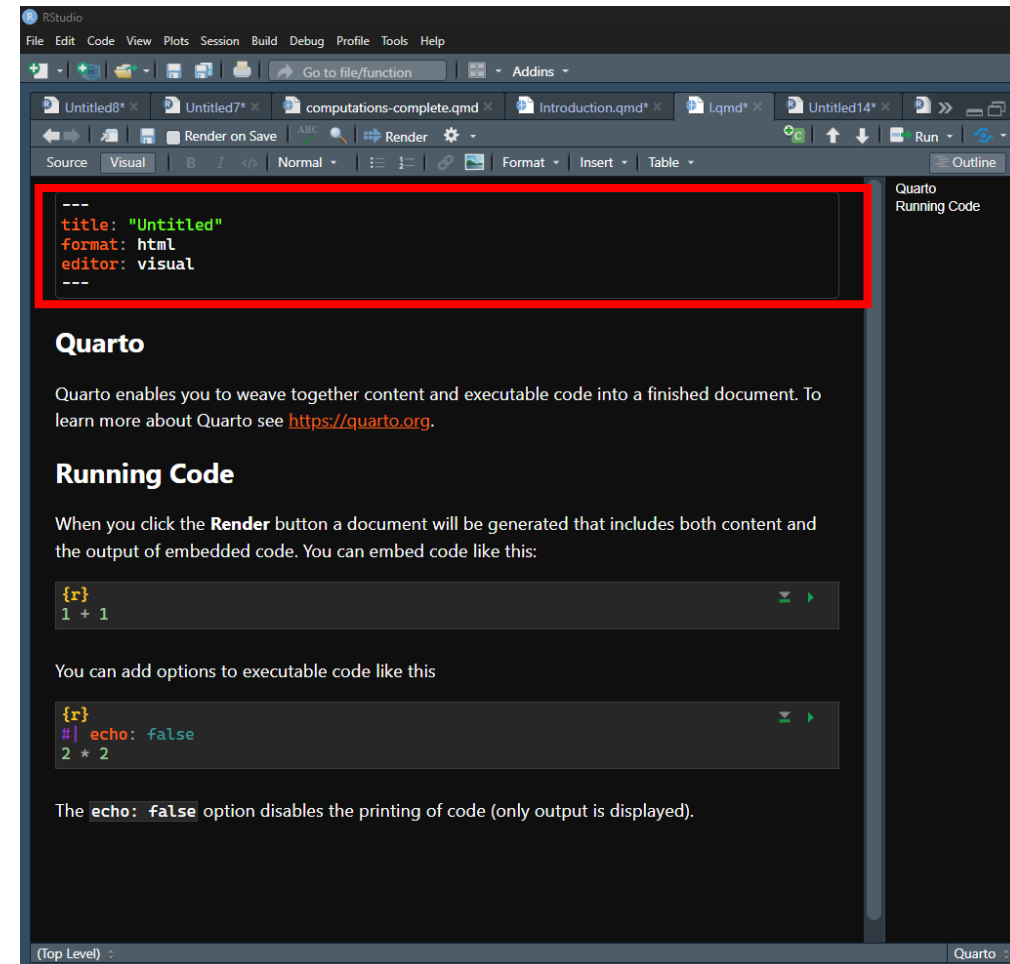
- Packages
- Working directory
- Running Code
- Plot results

The Outline panel is currently empty, indicating that no packages, working directory, or plot results have been generated yet. The status bar at the bottom shows the time 71:16 and the current page is Plot results.

... stay in *Visual*



# 2 - Output Settings



The screenshot shows the RStudio interface with a code chunk highlighted in red. The code chunk contains the following Quarto options:

```
---  
title: "Untitled"  
format: html  
editor: visual  
---
```

Below the code chunk, the Quarto documentation is visible, including sections for "Quarto" and "Running Code".

## 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).

## Syntax (Input)

```
---  
title: Introduction to Quarto  
subtitle: My subtitle  
author: Klajdi Puka  
date: last-modified  
format:  
  html:  
    self-contained: true  
execute:  
  echo: true  
  warning: false  
toc: true  
number-sections: true  
editor_options:  
  chunk_output_type: console  
---
```

## Explanation

`format` specifies the type of output file to generate. Here it is an HTML file.

`self-contained: true` specifies that the HTML file generated should be standalone file.

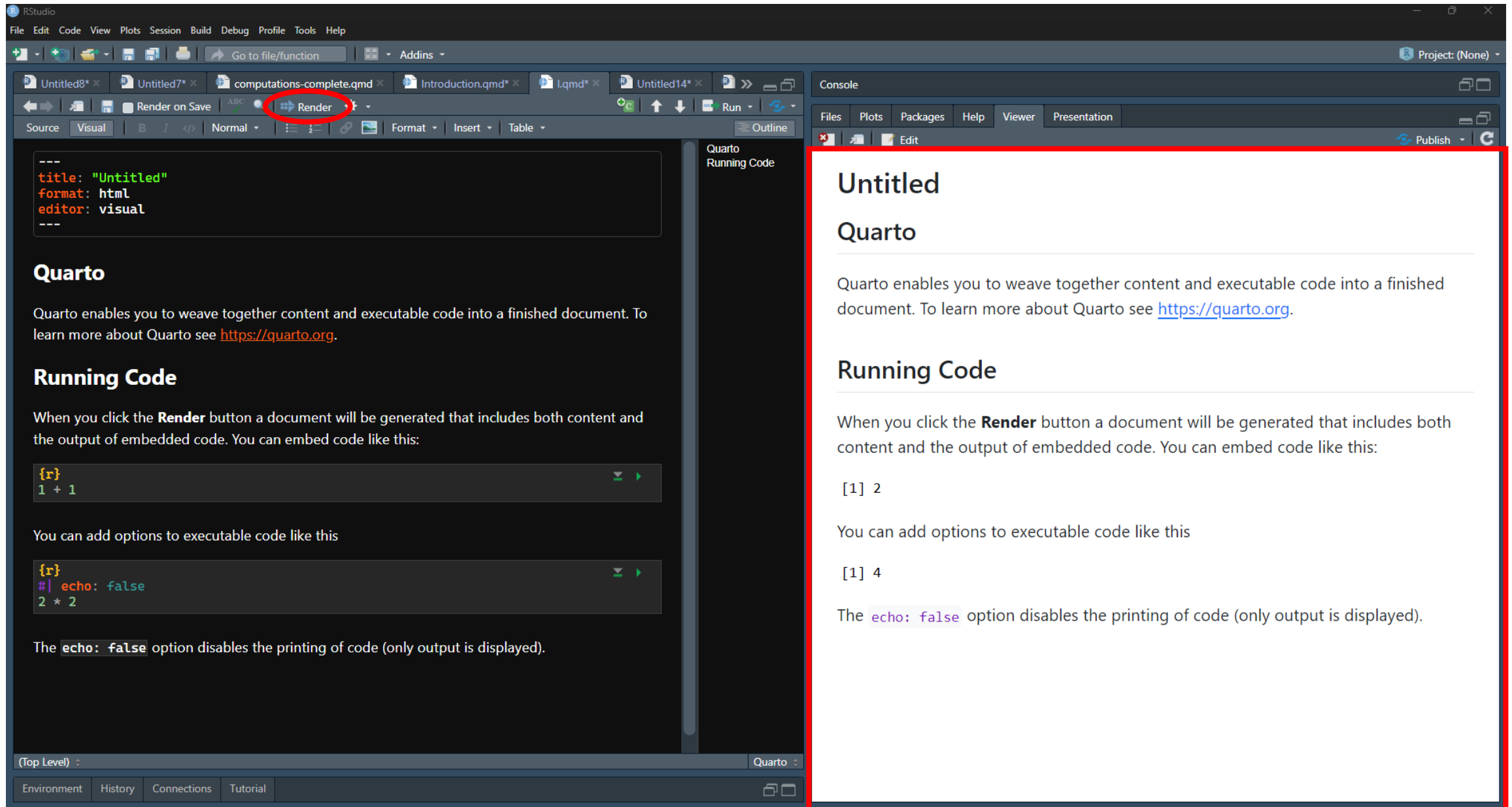
`echo: true` enables the printing of code (only output is displayed), unless otherwise specified.

`warning: false` disables the printing of warning messages.

`toc: true` specifies that the table of contents should be shown; automatically generated based on the headings.

`chunk_output_type: console` specifies that when executing the code in RStudio (i.e., when you “run” the code, instead of “render”), the output in RStudio should be displayed in the “console” (where it typically appears when using R)

# 2 - Output Settings – Render



The screenshot shows the RStudio interface. The top toolbar has a red circle around the **Render** button. The source editor on the left contains a Quarto document with the following content:

```
---  
title: "Untitled"  
format: html  
editor: visual  
---
```

The rendered output in the Console pane is as follows:

## Untitled

### 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

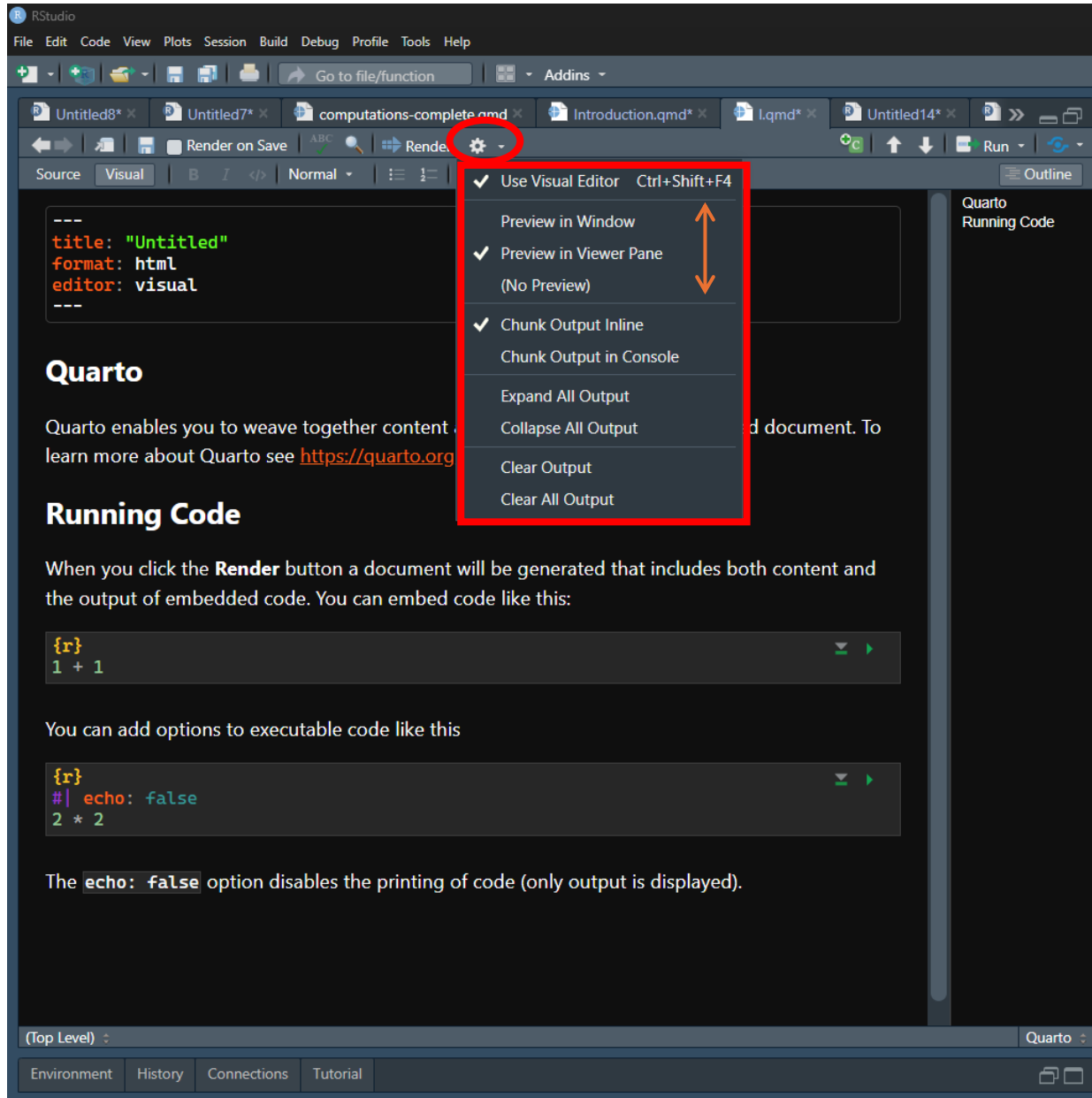
```
{r}  
#| echo: false  
2 * 2
```

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

The rendered output in the Console pane shows the following:

```
[1] 2  
  
[1] 4
```

## 2 - Output Settings – Render parameter



- **Preview in viewer pane:** output in R pane
- **Preview in window:** output open with your browser (html), Microsoft office (word), adobe reader (pdf), ...

## 2 - Output Settings – Exercice

(1) Change your render parameter to “*Preview in window*” and see what happen.

(2) Add “*echo: FALSE*” in your Output settings and see what happen.

(3) Type below  $1 + 1$ ,  $\log(-1)$ .  
add *warning: false* in your output settings.

(4) Use “html” and “pdf” as format: add

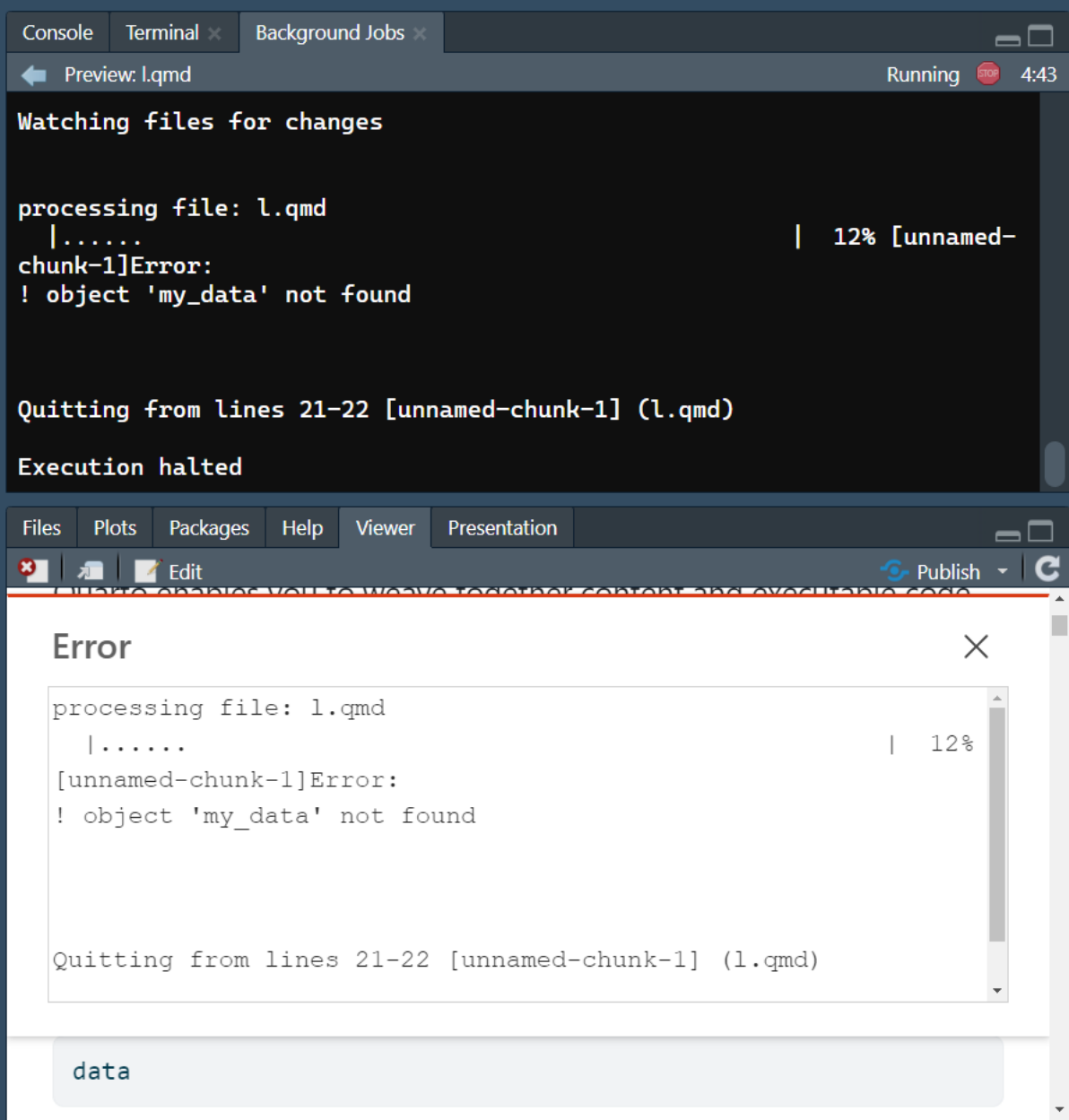
*html: default*

*pdf: default*

*docx: default*

And render to a pdf

## 2 - Output Settings – Error



The image shows a JupyterLab interface with a console window and an error dialog box. The console window is titled "Preview: l.qmd" and shows the following output:

```
Watching files for changes

processing file: l.qmd
|.....| 12% [unnamed-
chunk-1]Error:
! object 'my_data' not found

Quitting from lines 21-22 [unnamed-chunk-1] (l.qmd)

Execution halted
```

The error dialog box is titled "Error" and contains the same error message:

```
processing file: l.qmd
|.....| 12%
[unnamed-chunk-1]Error:
! object 'my_data' not found

Quitting from lines 21-22 [unnamed-chunk-1] (l.qmd)
```

At the bottom of the dialog box, there is a text input field containing the word "data".

# 3 - Formatting Basics

Syntax (Input)

```
# Heading 1

## Heading 2

### Heading 3

Bold text like **this** or __this__

Italicize text like *this* or _this_

superscript^2^ / subscript~2~

~strikethrough~
```



Formatted Output

## Heading 1

### Heading 2

---

#### Heading 3

Bold text like **this** or **this**

Italicize text like *this* or *this*

---

superscript<sup>2</sup> / subscript<sub>2</sub>

---

~~strikethrough~~

... still in *Visual* ?

# 3 - Formatting Basics – Exercise

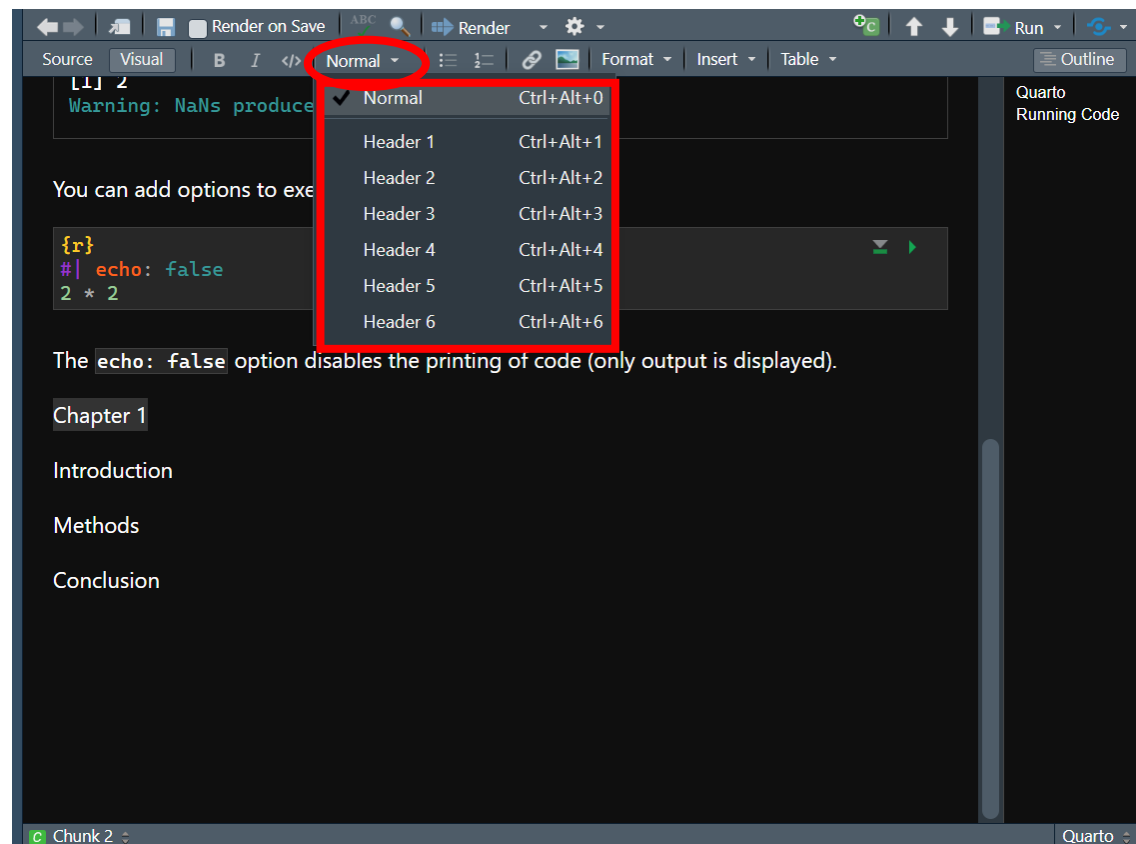
Add in your document:

Chapter 1 as *Header 1*

Introduction as *Header 2*

Conclusion as *Header 2*

With buttons



A screenshot of the RStudio interface. The 'Format' menu is open, showing options for text formatting. The 'Normal' option is selected and highlighted with a red box. Below it, a list of header options is shown, also highlighted with a red box:

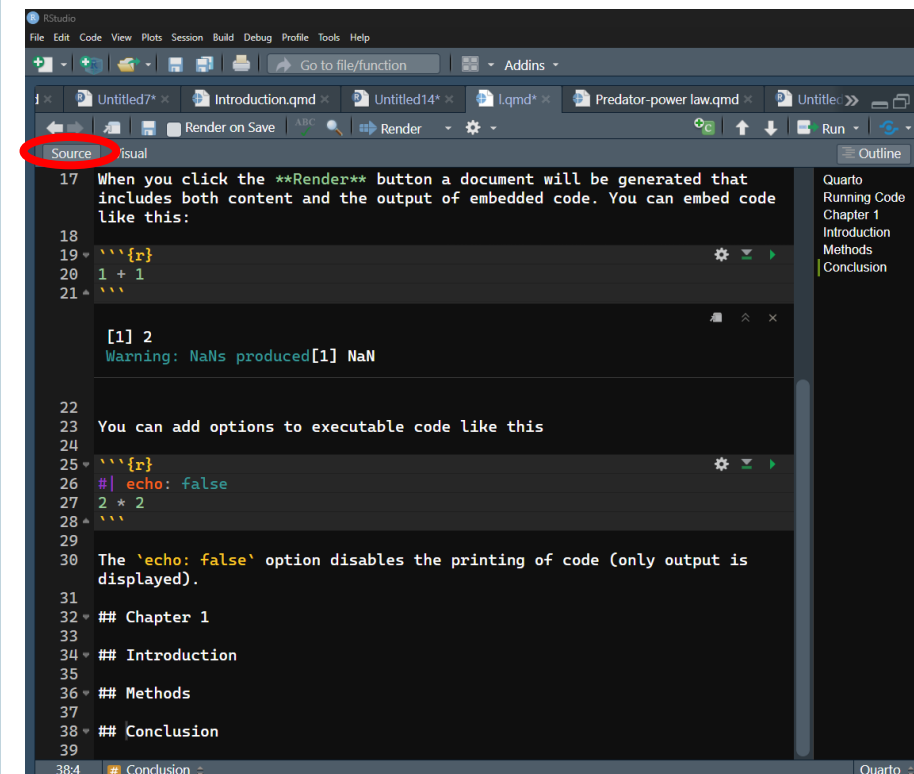
Option	Shortcut
Normal	Ctrl+Alt+0
Header 1	Ctrl+Alt+1
Header 2	Ctrl+Alt+2
Header 3	Ctrl+Alt+3
Header 4	Ctrl+Alt+4
Header 5	Ctrl+Alt+5
Header 6	Ctrl+Alt+6

The background shows a Quarto document with a code chunk containing `{r}`, `#| echo: false`, and `2 * 2`. The output shows a warning: `Warning: NaNs produced`. The document content includes a heading 'Chapter 1' and a paragraph 'You can add options to executable code like this'.

Manually

```
## Chapter 1
## Introduction
## Methods
## Conclusion
```

Source



A screenshot of the RStudio source editor. The code is as follows:

```
17 When you click the Render button a document will be generated that
18 includes both content and the output of embedded code. You can embed code
19 like this:
20 {r}
21 1 + 1
22
23 [1] 2
24 Warning: NaNs produced[1] NaN
25
26 You can add options to executable code like this
27
28 {r}
29 #| echo: false
30 2 * 2
31
32 The 'echo: false' option disables the printing of code (only output is
33 displayed).
34
35 ## Chapter 1
36
37 ## Introduction
38
39 ## Methods
40
41 ## Conclusion
```

The output of the code chunks is visible in the console area on the right, showing the warning message and the result of the calculations.

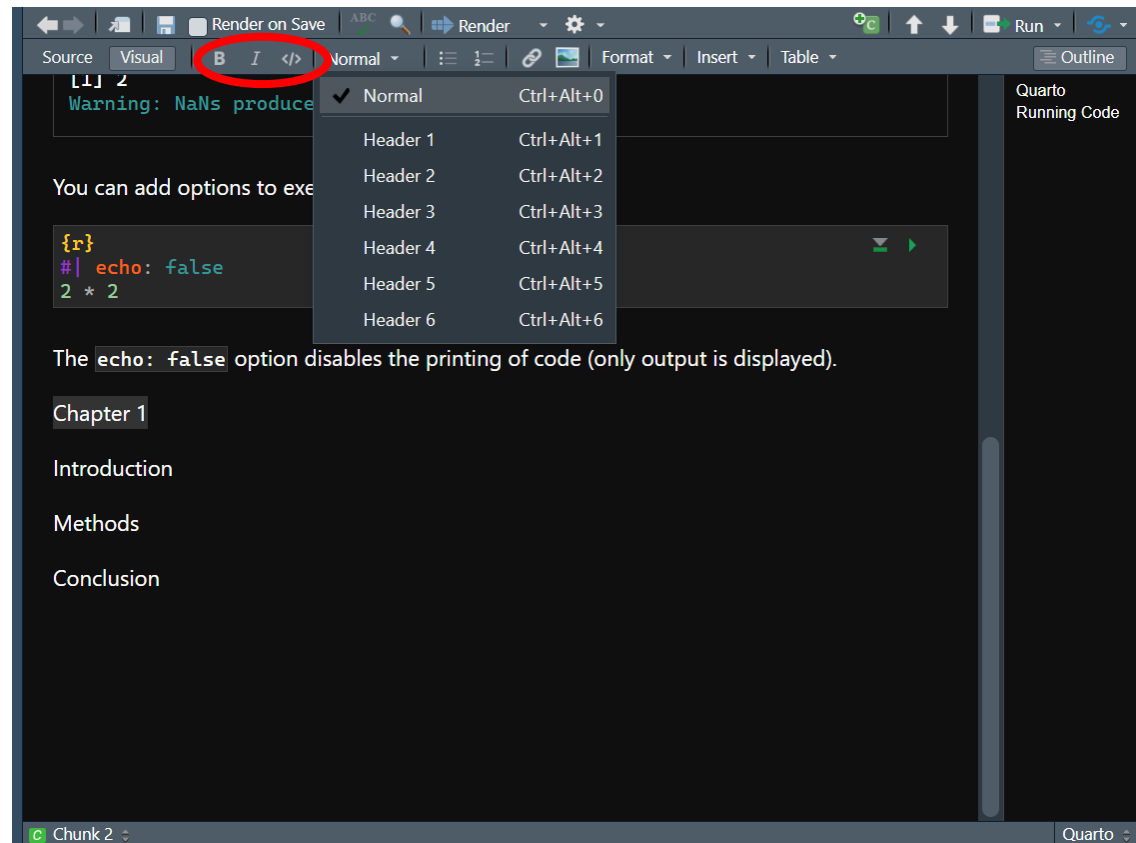


# 3 - Formatting Basics – Exercise

Add in your document:

This one is **bold**, but this one is *italic*. I am ***bolditalic***.

With buttons



The screenshot shows the RStudio interface. The 'Visual' menu is open, and the 'B' (Bold) and 'I' (Italic) buttons are circled in red. Below the menu, there is a code chunk with the following content:

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

The output of the code chunk is displayed as:

```
[1] 2
Warning: NaNs produced
```

Below the code and output, there is a text block:

You can add options to executable code like this

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

Chapter 1

Introduction

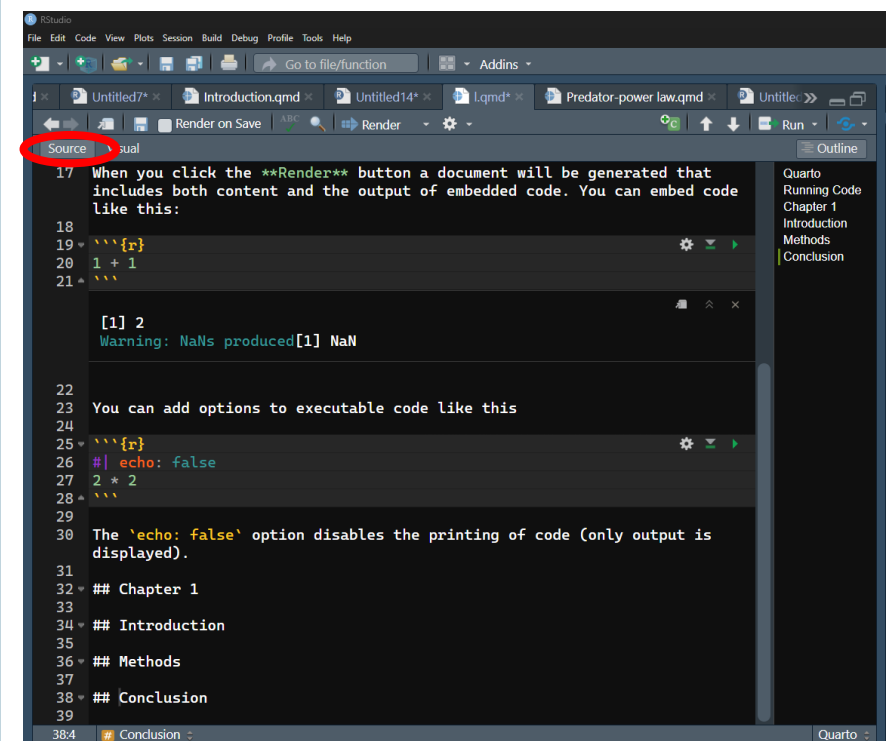
Methods

Conclusion

Manually

```
## Chapter 1
## Introduction
## Methods
## Conclusion
```

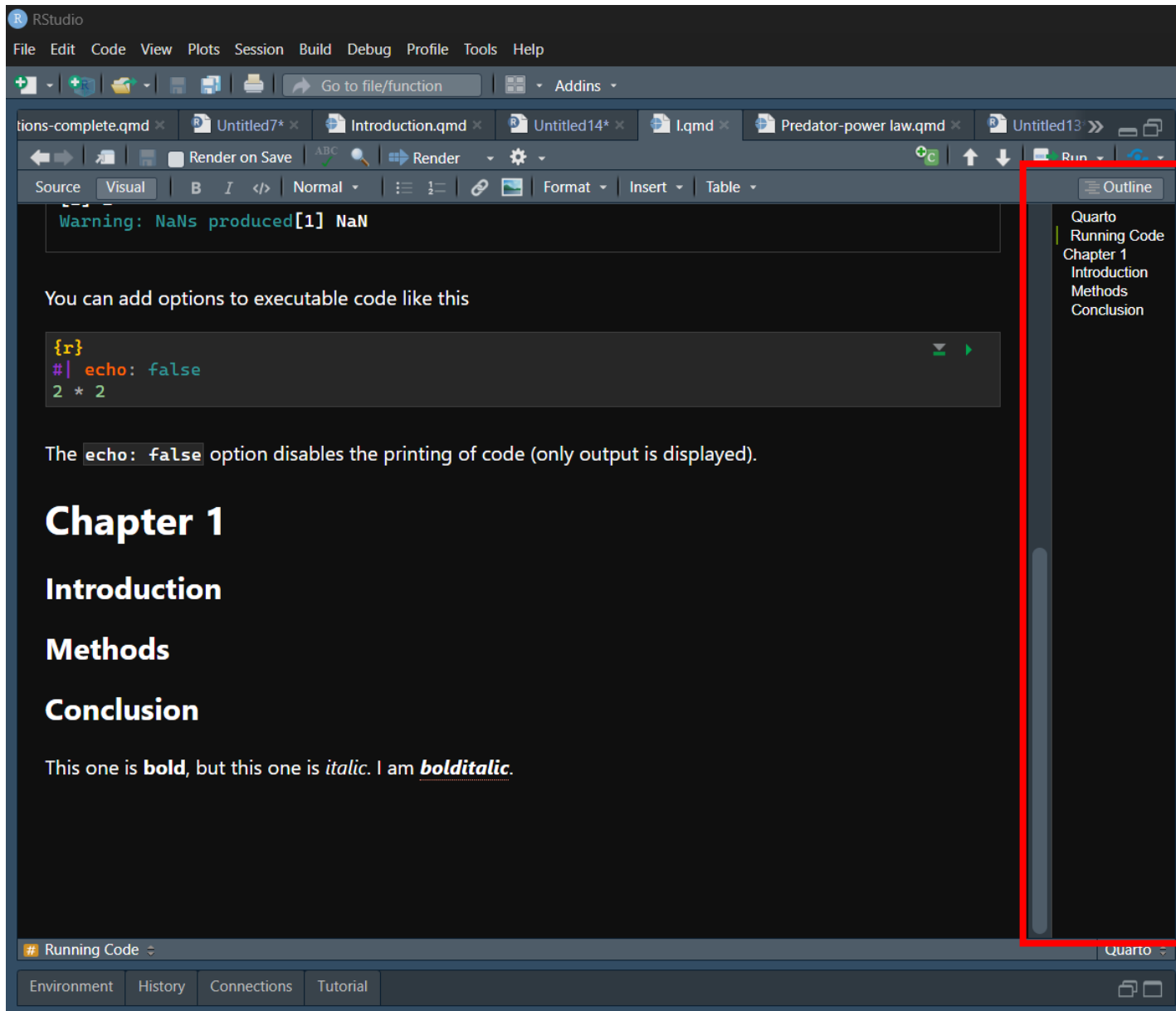
Source



The screenshot shows the RStudio interface with the 'Source' button circled in red. The source code is as follows:

```
17 When you click the **Render** button a document will be generated that
18 includes both content and the output of embedded code. You can embed code
19 like this:
20 ```{r}
21 1 + 1
22 ```
23
24 [1] 2
25 Warning: NaNs produced[1] NaN
26
27 You can add options to executable code like this
28
29 ```{r}
30 #| echo: false
31 2 * 2
32 ```
33
34 The 'echo: false' option disables the printing of code (only output is
35 displayed).
36
37 ## Chapter 1
38 ## Introduction
39 ## Methods
40 ## Conclusion
```

# 3 - Formatting Basics – Exercise



The screenshot shows the RStudio interface with a Quarto document open. The main editor displays a warning about NaNs, followed by a code block with `echo: false` and `2 * 2`. Below the code, there is a paragraph explaining the `echo: false` option. The document content includes a table of contents with the following items:

- Chapter 1
  - Introduction
  - Methods
  - Conclusion

The bottom of the editor shows the text: "This one is **bold**, but this one is *italic*. I am ***bolditalic***." The right-hand pane, titled "Outline", is highlighted with a red box and contains the same table of contents structure.

- Headers are displayed in outline

# 3 - Formatting Basics

- The following syntax works with Quarto and its predecessor Markdown. In addition to this code, RStudio has a 'visual' editor which provides a point-and-click interface to format the document.

- Bullet point (nordered lists)
- Hyphen, followed by 'tab'

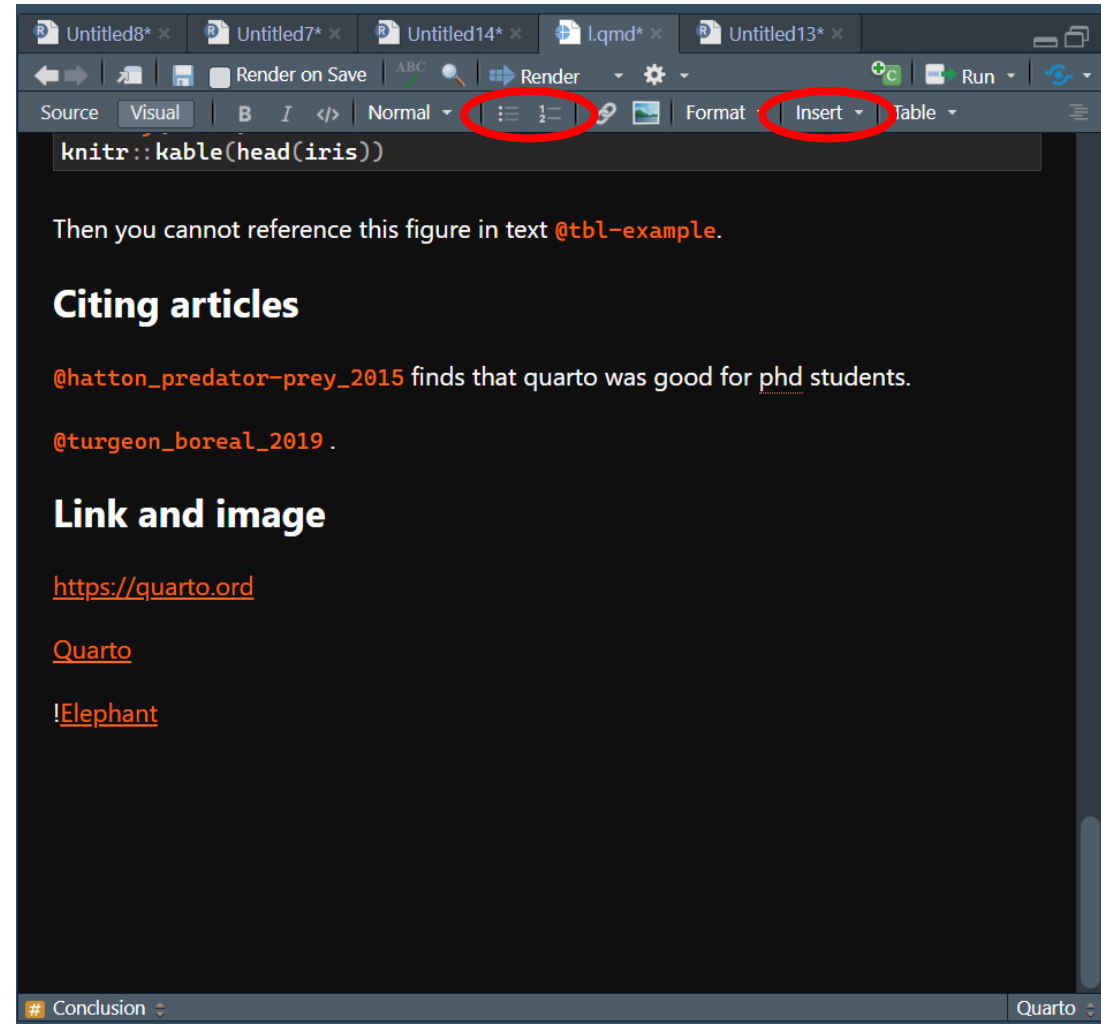
1. Ordered list
2. Number, period, then 'tab'

	Manual	Table
Variable 1	11	21
Variable 2	12	22
Variable 3	12	23

- Bullet point (nordered lists)
- Hyphen, followed by 'tab'

1. Ordered list
2. Number, period, then 'tab'

	Manual	Table
Variable 1	11	21
Variable 2	12	22
Variable 3	12	23



# Technical terms

- **Inline Code:** allows you to execute code within the text.
- **Outline Code:** allows you to execute code in chunks.

# 4 - Integrating R Statistical Code – chunks

To execute R code

- **Outline**

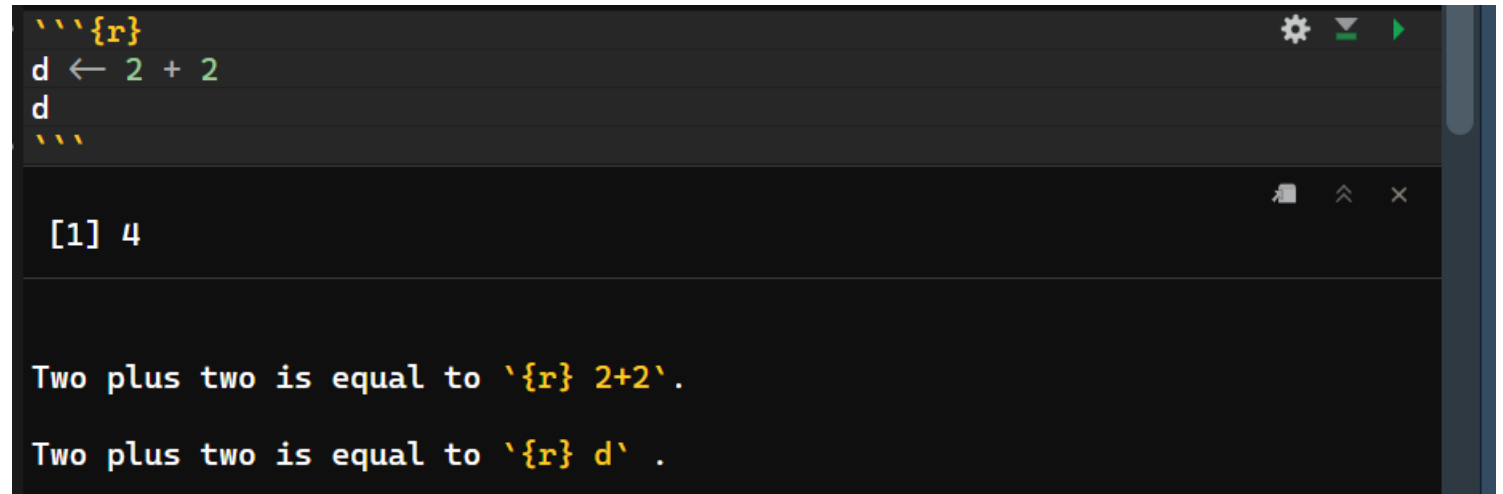
```
```${r}```
```

Shortcut: **Crtl + Alt + I**

- **Inline:**

```
`{r} ...`
```

< Source



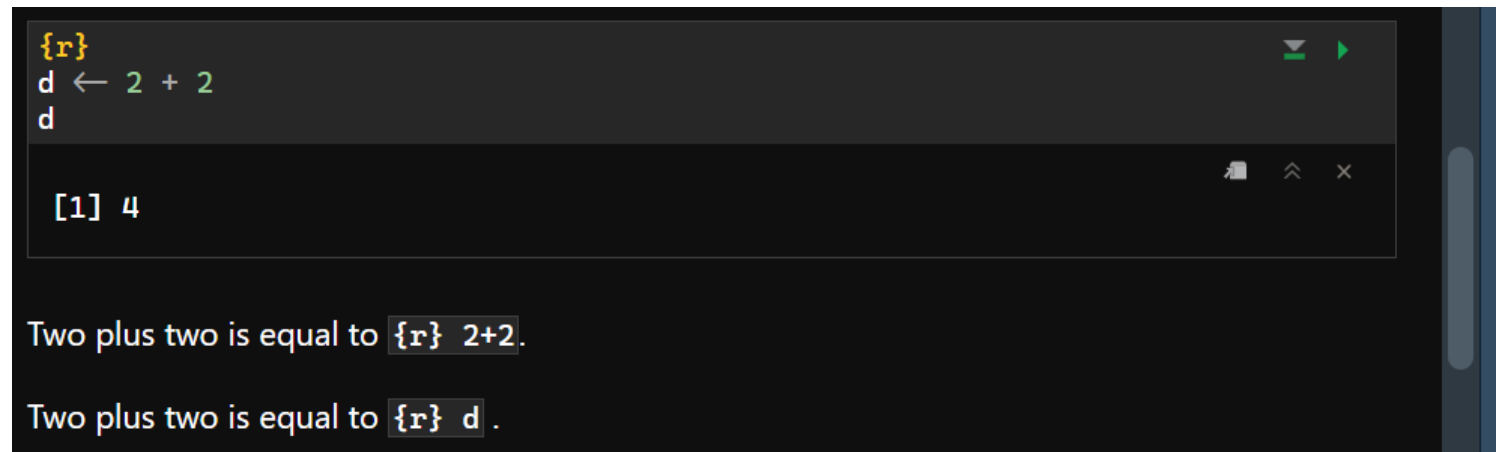
```
```${r}```  
d ← 2 + 2  
d  
```${r}```
```

[1] 4

Two plus two is equal to `{r} 2+2`.

Two plus two is equal to `{r} d`.

< Visual



```
{r}  
d ← 2 + 2  
d
```

[1] 4

Two plus two is equal to `{r} 2+2`.

Two plus two is equal to `{r} d`.

## 4 - Integrating R Statistical Code – Exercice

(1) Create an object  $d = 2 + 2$  in a chunk

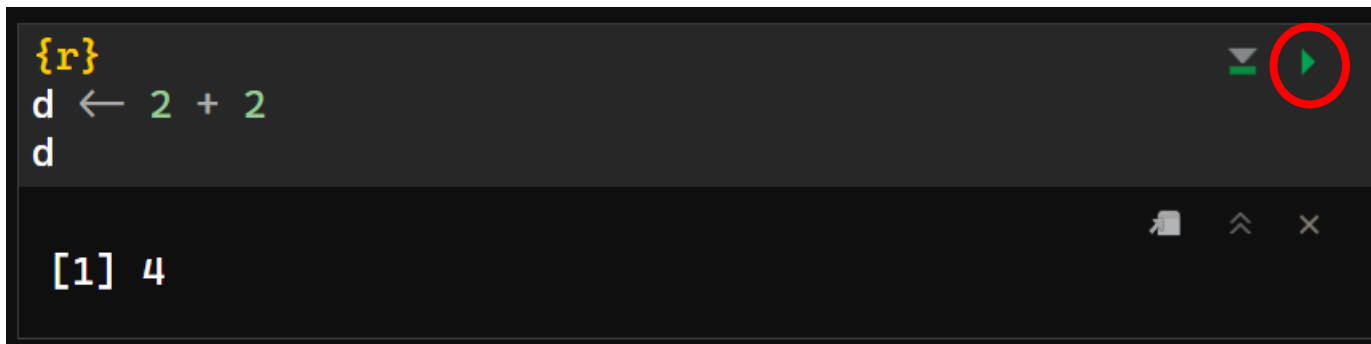
```
`{r}  
d ← 2+2  
`
```

(2) In the body text, type:

Two plus two is equal to `{r} 2+2``

Two plus two is equal to `{r} d``

Display your R codes without render ...



```
{r}  
d ← 2 + 2  
d  
[1] 4
```

## 4 - Integrating R Statistical Code – Execution Options

Allow to handle chunks output ... use `#|`

```
{r}
#| echo: false
#| warning: false
```

Other execution options include:

- `eval: true` indicates that the code chunk should be ‘evaluated’ or ‘run’
- `echo: true` indicates the code chunk should be ‘echoed’ or shown along in the output
- `warning: false` indicates that any warning messages should not be shown in the output
- `output: asis` indicates that the output is raw markdown and should not have any of Quarto’s standard enclosing markdown
- `include: false` is a catch all, indicating neither the code chunk nor results should be shown in the output
- `file: "Name of R file.R"` can be used to import, display, and run the code that is in a separate file called “Name of R file.R”. Useful if you have custom functions saved in a separate R script, and want to show the function code in your final document.

## 4 - Integrating R Statistical Code – Exercice

- Type the following code in your document and see how is the output by clicking on render.

```
{r}
#| include: false
plot(1:10, 1:10)
```

```
{r}
plot(1:10, 1:10, col="red")
```

```
{r}
#| echo: false
plot(1:10, 1:10, col="blue")
```

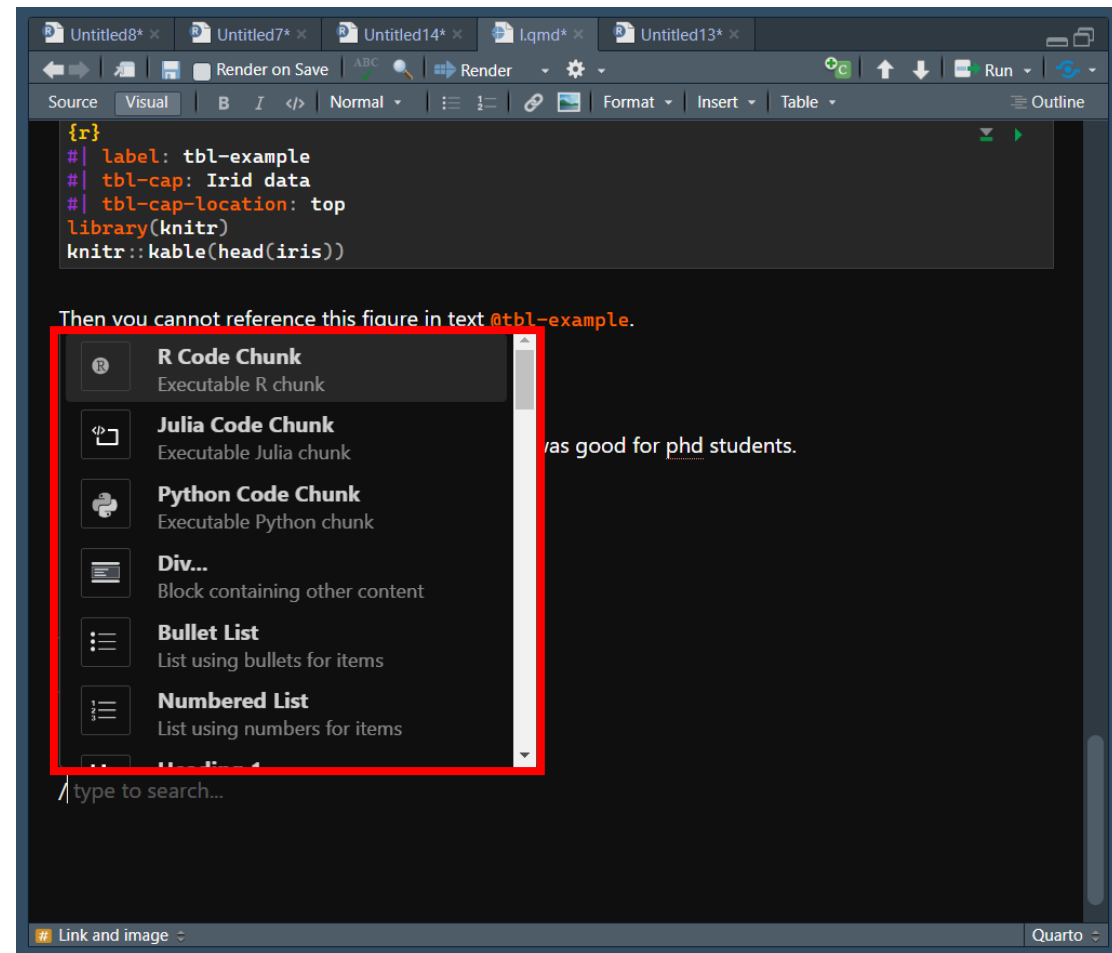


# 4 - Integrating R Statistical Code - **workflow**

- Package codes, working directory, ... should be put in R chunks.
- Objects already created in a chunk can be used in other chunks in the documents ...

## Shortcut / display options

- Type `/...`



# 5 - Technical writing - Equation

- Use  $\$ \dots \$$  for inline
- For outline, use

$\$\$$

$\dots$

$\$\$$

• Einstein equation:  $\$E = mc^2\$$   $\rightarrow$  Einstein's equation:  $E = mc^2$

•  $\$\$$   
 $E = mc^2$   $\rightarrow$   $E = mc^2$   
 $\$\$$

•  $\$\$$   
 $\frac{1}{2}$   $\rightarrow$   $\frac{1}{2}$   
 $\$\$$

•  $\$\$$   
 $Y = X_1 + X_2$   $\rightarrow$   $Y = X_1 + X_2$   
 $\$\$$

•  $\$\$$   
 $\sqrt{p}$   $\rightarrow$   $\sqrt{p}$   
 $\$\$$

•  $\$\$$   
 $\sum_{i=1}^n (\bar{x} - x_i)^2$   $\rightarrow$   $\sum_{i=1}^n (\bar{x} - x_i)^2$   
 $\$\$$

•  $\$\$$   
 $Y \sim X\beta_0 + X\beta_1 + \epsilon$   $\rightarrow$   $Y \sim X\beta_0 + X\beta_1 + \epsilon$   
 $\$\$$

•  $\$\$$   
 $\Pr(\theta | y) = \frac{\Pr(y | \theta)\Pr(\theta)}{\Pr(y)}$   $\rightarrow$   $\Pr(\theta | y) = \frac{\Pr(y | \theta)\Pr(\theta)}{\Pr(y)}$   
 $\$\$$

# 6 - Referencing: citing Figures, Tables, links, ...

Link to tables or figures or section in text.

- Step 1: create the figure/image in a chunk

- Step 2: add the execution option `#| label:`

`#| label: fig-label`

`#| label: tbl-label`

- Step 3: refer it in your text using `@...`

`@fig-label`

`@tbl-label`

# Referencing figures

- Outline (chunks)

```
{r}
#| label: fig-example
#| fig-cap: Volcano heatmap
image(volcano)
```

Then you cannot reference this figure in text `@fig-example`.

- Inline

Type	Syntax	Output
Default	<code>@fig-elephant</code>	Figure 1
Capitalized	<code>@Fig-elephant</code>	Figure 1
Custom Prefix	<code>[Fig @fig-elephant]</code>	Fig 1
No Prefix	<code>[-@fig-elephant]</code>	1

# Referencing tables

- Created in Outline (chunks)

```
{r}
#| label: tbl-example
#| tbl-cap: Irid data
#| tbl-cap-location: top
library(knitr)
knitr::kable(head(iris))
```

Then you cannot reference this figure in text `@tbl-example`.

- Used in Inline

Type	Syntax	Output
Default	<code>@fig-elephant</code>	Figure 1
Capitalized	<code>@Fig-elephant</code>	Figure 1
Custom Prefix	<code>[Fig @fig-elephant]</code>	Fig 1
No Prefix	<code>[-@fig-elephant]</code>	1

# Citing articles

(1) Exporter votre library (ex: Zotero, Mendley, ...) en **.bib**






(2) Add the file in your working directory.

(3) Add **bibliography: name.bib** in the output settings.

```
---  
title: "Untitled"  
format:  
  html: default  
  pdf: default  
editor: visual  
warning: false  
bibliography: Library.bib  
---
```

(4) In the text type @....

**Referencing tables**

-  **@turgeon\_boreal\_2019** Turgeon, K 2019  
Boreal river impoundments caused nearshore fish commu...
-  **@turgeon\_dams\_2018** Turgeon, K 2018  
Dams have varying impacts on fish communities across la...
-  **@turgeon\_empirical\_2021** Turgeon, K 2021  
Empirical characterization factors to be used in LCA and a...
-  **@turgeon\_fish\_2015** Turgeon, K 2015  
Fish population dynamics and diversity in boreal and tem...
-  **@turgeon\_novel\_2016** Turgeon, K 2016  
Do novel ecosystems follow predictable trajectories? Testi...

@turg ←

phd students.

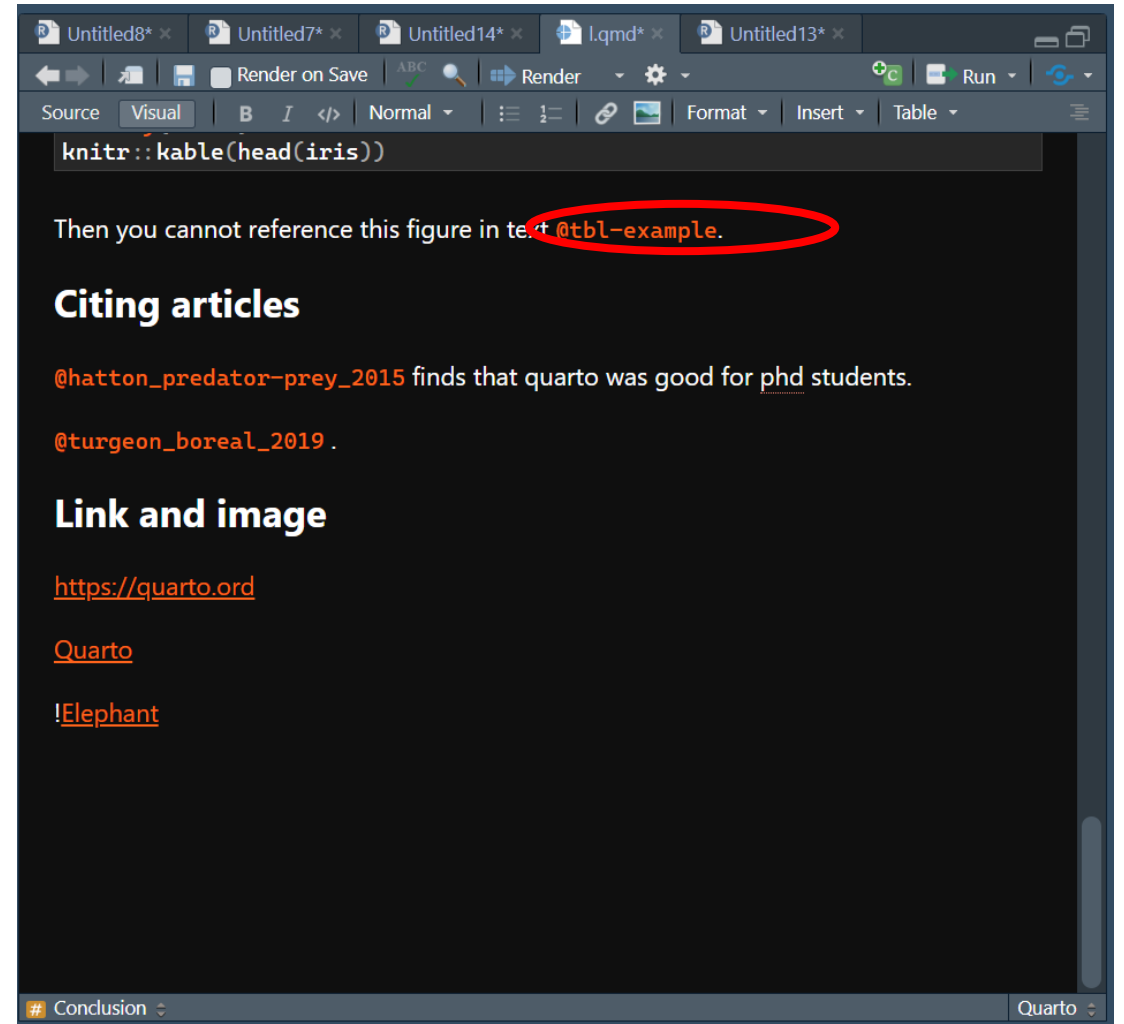
# Referencing link, image, ...

Clickable link: `<https://google.com>`

`[Hyperlink](google.com)`

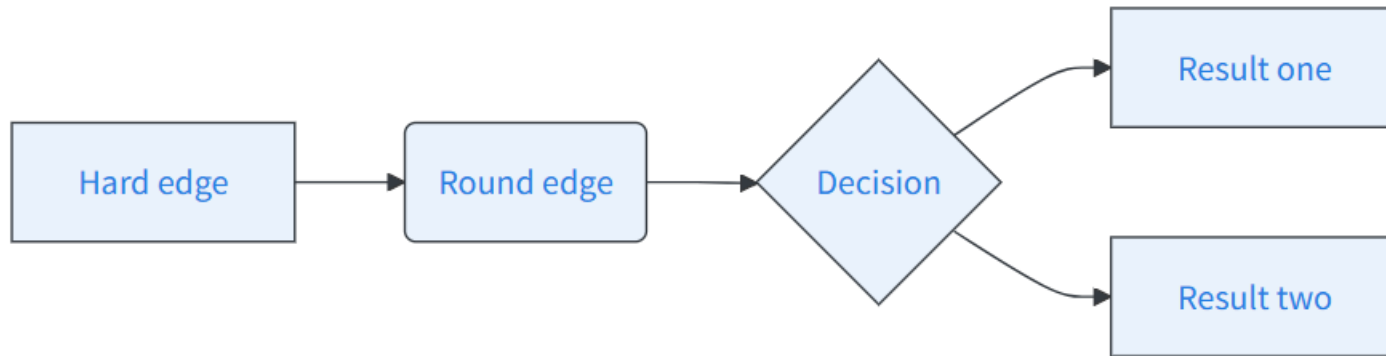
Clickable link: <https://google.com>

[Hyperlink](#)



# Diagrams

```
```{mermaid}
flowchart LR
  A[Hard edge] --> B(Round edge)
  B --> C{Decision}
  C --> D[Result one]
  C --> E[Result two]
```
```





# Common shortcuts

| Action                           | Windows/Linux        | Mac                        |
|----------------------------------|----------------------|----------------------------|
| Knit document                    | Ctrl + Shift + K     | Cmd + Shift + K            |
| Insert Chunk                     | Ctrl + Alt + I       | Cmd + Option + I           |
| Run Current Chunk                | Ctrl + Alt + C       | Cmd + Option + C           |
| Jump to                          | Shift+Alt+J          | Cmd+Shift+Option+J         |
| Show Keyboard Shortcut Reference | Alt+Shift+K          | Option+Shift+K             |
| Create multiple cursors          | Ctrl + Alt + Up/Down | option + control + Up/Down |
| Delete the current line          | Ctrl + D             | Cmd + D                    |
| Un/Comment out a line            | Ctrl + Shift + C     | Cmd + Shift + C            |
| Reformat Section                 | Ctrl + Shift + A     | Cmd + Shift + A            |

The end ...