

Lab 1: Introduction to Quarto

Qian Wu

2025 Fall Session 2

[click here for latest version](#)

1 What is Quarto?

Quarto is a document container that allows you to mix code, output, and text to produce reproducible results and auto-generate high quality document.

2 Install Quarto

To install Quarto, please enter the following line of code in the Terminal window (not the Console window):

```
quarto install tinytex
```



A screenshot of a terminal window. The window has three tabs at the top: "Console", "Terminal" (which is highlighted with a red oval), and "Background Jobs". Below the tabs, there are two small blue arrows pointing left and right. To the right of the arrows is the text "Terminal 1". To the right of that is a dropdown arrow pointing down. To the right of the dropdown arrow is the path "/Github/dkustats101spr2025s4". At the bottom of the terminal window, the prompt "DKU-M272:dkustats101spr2025s4 awm31\$ quarto install tinytex" is visible. The entire terminal window is enclosed in a light gray border.

3 Practicing with Quarto

To complete this lab, please do a simple investigation into the distribution of **mpg** (miles per gallon) and **wt** (weight) from the **mtcars** built-in dataset.

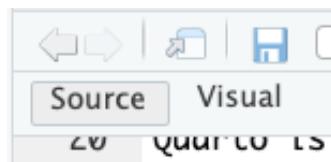
1. Select File → New File → Quarto Document.
2. Create YAML header.
3. Insert an R code block and then make a histogram of **mpg**.

4. Describe features of the mpg distribution (center, spread).
5. Repeat for the variable **wt**.
6. Add appropriate headers for each section and make sure you can navigate between them.
7. Write a brief conclusion of your investigation.

Feeling overwhelmed? Turn to next page for step-by-step instructions!

3.1 Two Views to Edit Document

RStudio has two views for editing your document, Source and Visual.



Most people find it easier to edit using the Visual mode. Switch back and forth between the two views to check the differences.

3.2 Create YAML Header

At the top of your document, there is a metadata block that controls the document's properties and appearance. This is the YAML header.

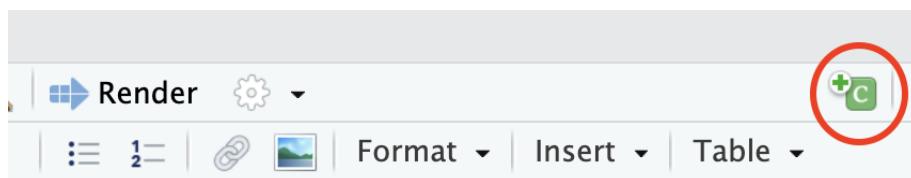
```
1 ---  
2 title: "Untitled"  
3 editor: visual  
4 ---  
5
```

You CANNOT put R code in this section and it must always be at the top of your document. For common YAML fields, check here.

[Please add a subtitle and an abstract to your YAML header.](#)

3.3 Insert R Code Block

To write R code, you need to add a code block with the “Insert a New Code Chunk” button.



3.3.1 Setup block

We often begin the R code with the setup block. In the setup block, you will set working directory, clean objectives, and library (load) all packages needed to run the code.¹ A good setup block looks like this:

```
13 - ``{r}
14  #| label: setup
15  #| include: false
16
17  library(tidyverse)
18 - ```
19
```

Please add a setup block to load the `tidyverse` library.

3.3.2 Other blocks

You might notice that, the setup bock contains “#|” symbols. In Quarto, the “#|” prefix in a code block is used for YAML option syntax within code cells. It allows you to specify cell-specific options directly in the code chunk. Each code block header starts with

```
#| <option>: <value>
```

For example, here is a simple code block to produce a histogram with `ggplot` function.

```
74 - ``{r}
75  #| label: test-hist
76  #| fig-cap: "Histogram of variable: MPG"
77  #| echo: true
78
79  ggplot(mtcars, aes(x=mpg)) +
80    geom_histogram()
81 - ````
```

What do the code block headers (line 75-77) do?

- “#| label: test-hist” labels this code block, indicating we are creating a test histogram.²
- “#| fig-cap: “Histogram of variable: MPG”” assigns the caption “Histogram of variable: MPG” to the output histogram.
- “#| echo: true” specifies both the R code and the output histogram (with caption) should appear in the rendered PDF.

¹Note: do not put install packages commands. You only need to install packages once.

²It's a good habit to label each code block so it is easy to move around your document and recall what each block does.

The remaining lines (line 79-80) use `ggplot` function to draw a histogram for the variable `mpg` in the dataset `mtcars`.

Please create two code blocks, one that produces a histogram of the variable `mpg` and the other that finds the mean and the standard deviation of `mpg` (you can use the following code example below). Label these two blocks and your setup block.

```
mtcars %>%
  summarise(mean_mpg = mean(mpg), sd_mpg = sd(mpg))
```

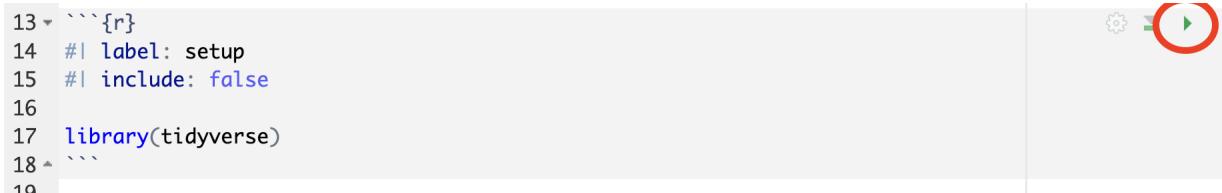
3.4 Execute Code

There are two ways to execute (run) your code.

- Execute entire script: The Render button renders your document from the very first block, going all the way to the end (unless there is an error, in which case it will stop).



- Execute each block: The green triangle on the right top corner executes only the code in that block. This can be useful to quickly check if your code works. However, it can be dangerous because (1) it can call back the objectives you have deleted the code for; (2) the preview window can display unreliable formats.



I consider it best practice to use the Render button regularly to make sure your document will work at submission time and display the way you prefer.

Please click Render button and verify your document will produce the desired outcome.

3.5 Label Graphs and Tables

Sometimes, we want to auto-number and cross-reference graphs and tables throughout a document.³ Quarto allows us to do these in a systematic way. For code blocks with figures, we can label figures by adding a block header:

```
#| label: fig-<your choice of label>
```

³See here and here for instructions on graphs and tables, respectively.

For example, “#| label: fig-mtcarsmpghist” can be a good label for the histogram of **mpg** in the dataset **mtcars**.

Replace the “#| label: test-hist” header with “#| label: fig-<your choice of label> to label your histogram.

When there are multiple graphs, you can assign different captions to them by adding a block header:

```
#| fig-subcap:  
#|   - "<caption for subgraph 1>"  
#|   - "<caption for subgraph 2>"
```

Following this page, in the test-hist block, add a second histogram of **mtcars\$mpg** but change the binwidth. Create subcaptions for the two plots.

3.6 Repeat for Variable **wt**

Now repeat the operations above for the variable **wt**: draw histogram and calculate mean and standard deviation. Write a few sentences interpreting your results.

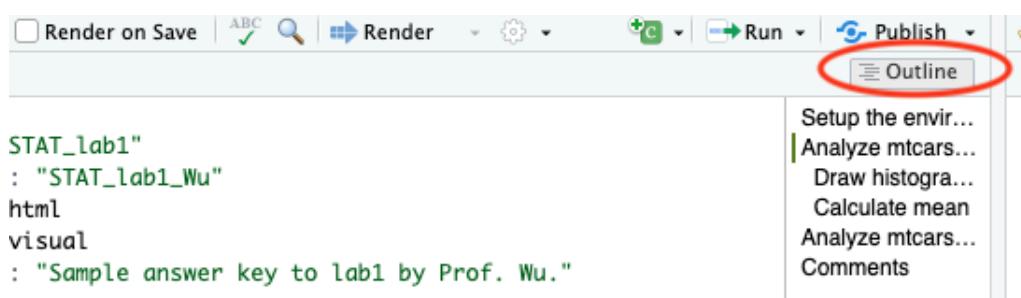
3.6.1 Navigate document

To make your document navigable (auto-generate a content), you need to specify headings for your document sections. We use “#” for sections, “##” for subsections, and so on so forth.

```
17  
18 # Analyze mtcars$mpg  
19 ## Draw histograms  
20 {r}  
21 #| label: test-hist  
22 #| fig-cap: "Histogram of MPG"  
23 #| label: fig-mtcarsmpghist  
24 #| echo: true  
25  
26 ggplot(mtcars, aes(x=mpg)) +  
27   geom_histogram()  
28 ...
```

You can navigate your document in two ways.

- Outline window on the top right corner.



- Menu list on the bottom left corner.

```
11 #| STAT_lab1
12 #| Setup the environment
13
14 li Chunk 1: setup
15
16 `` Analyze mtcars$mpg
17
18 #| Draw histograms
19 #| Chunk 2: test-hist
20 #| Calculate mean
21 #| Chunk 3: calculate mean for mpg
22 #| Analyze mtcars$wt
23 #| Comments
```

18:21 # Analyze mtcars\$mpg

4 Finish up

Once you finish, please show me your result and then you can proceed to this website for a computations tutorial. Please try to modify your document according to the exercises on the tutorial webpage until the end of class.