Audio Connection and Chat (Zoom)

NIH Library
Office of Research Services
Serving the NIH Community

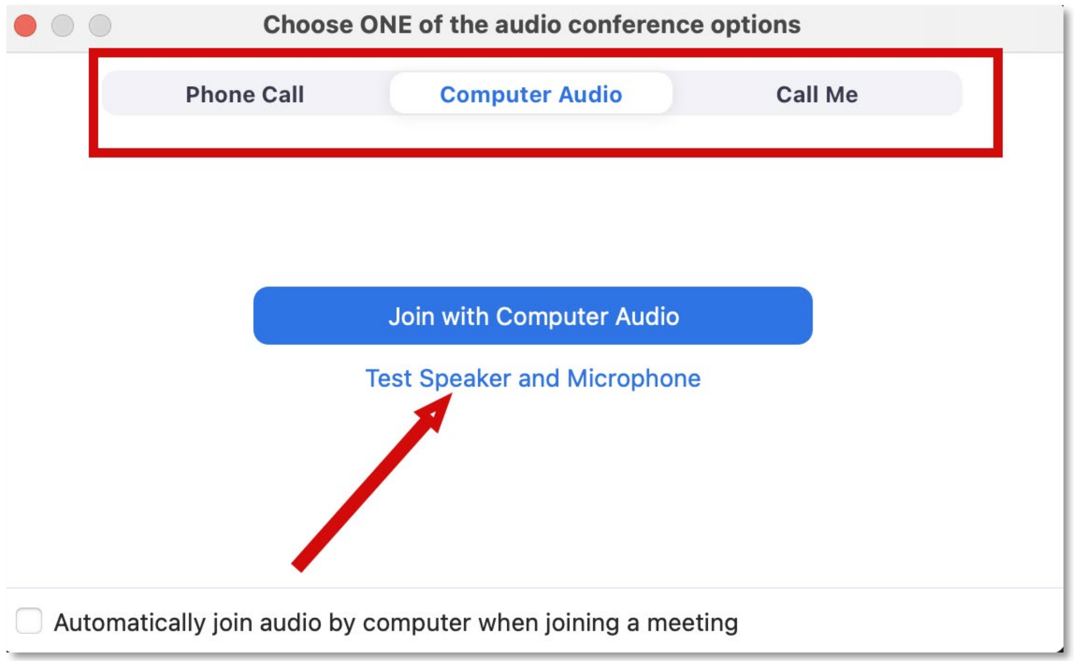
You will not hear any sound until the webinar starts.

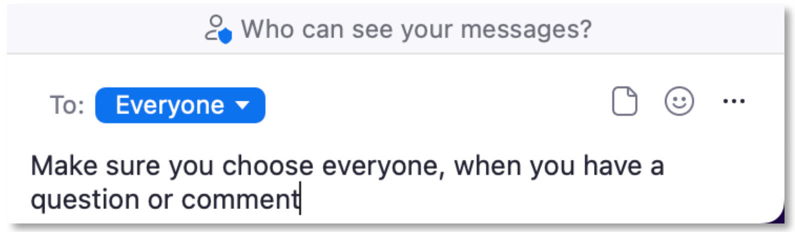
Connect Audio

1. When you join Zoom, the ***Join Audio*** preferences box pops-up (Phone Call, Computer Audio, or Call Me)
2. Choose an option that works best for you
3. Join using that option
4. Use Test Speakers and Microphone option to optimize your webinar experience

Chat

Please send your chat to ***Everyone*** to make sure the monitor sees your question


Choose ONE of the audio conference options
Phone Call
Computer Audio
Call Me
Join with Computer Audio
Test Speaker and Microphone
Automatically join audio by computer when joining a meeting



Make sure you choose everyone, when you have a question or comment

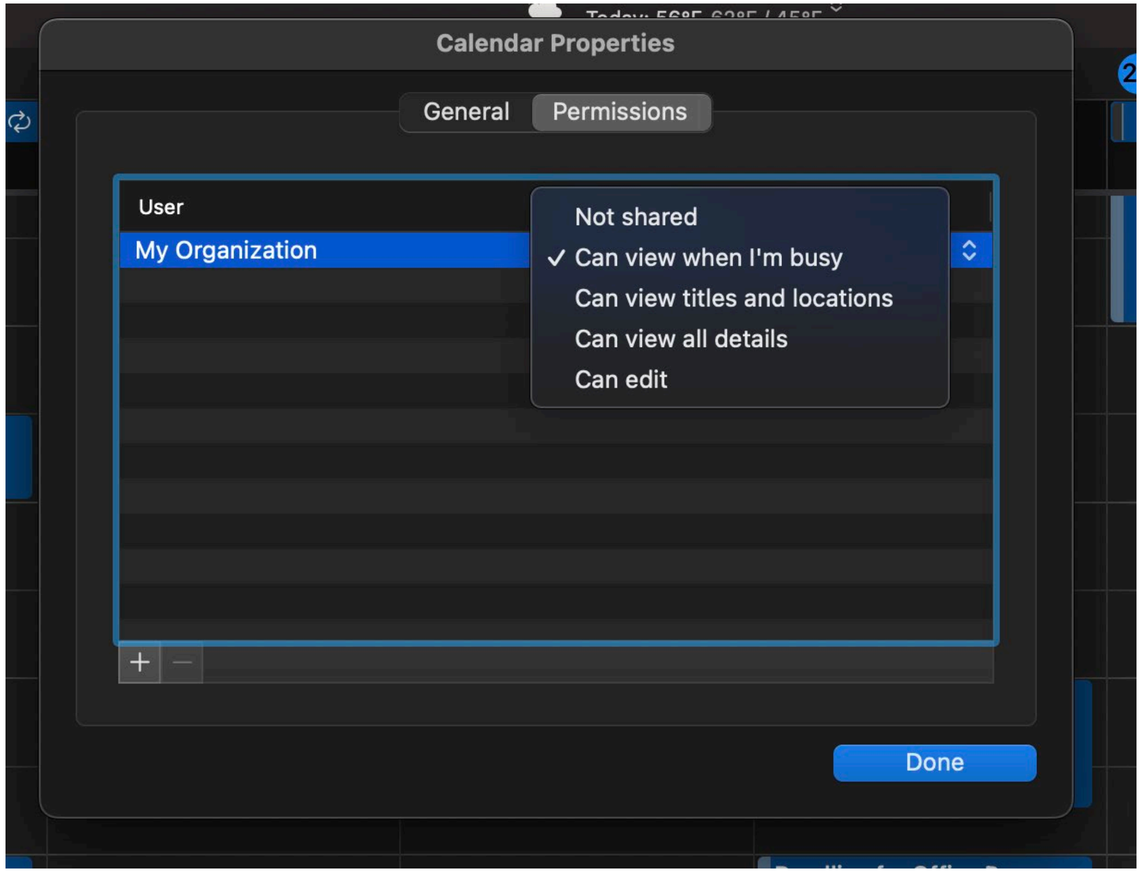
NIH National Institutes of Health
Office of Management

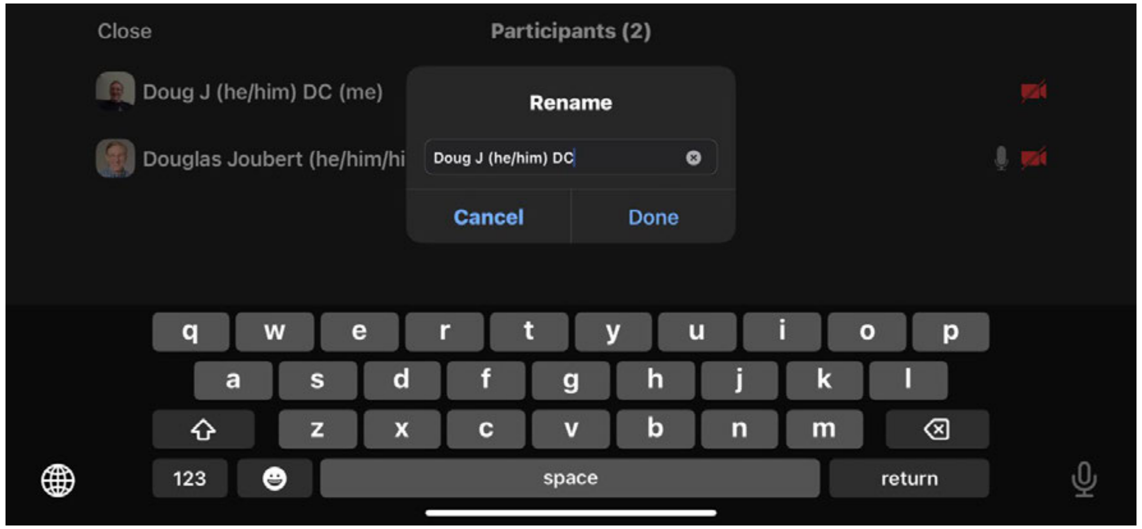
Reproducibility in RStudio: Basic Markdown

Doug Joubert

2023-03-07

Resources from PowerPoint





Please rename yourself, so we can:

- Send you the student version of the PowerPoint
- Send your training certificate
- Add you to our list-serve

Class Description

- Designed for those who want to extend the basics of R Markdown and apply those skills in Quarto.
- Quarto is an open-source scientific and technical publishing system that offers multilingual programming language support
- You will learn about the similarities and differences between R-markdown and Quarto
- You will also learn how to use Quarto to render documents in multiple formats, with a focus on scholarly publishing

Class Objectives

- Distinguish between R-markdown and Quarto
- Demonstrated the difference between the visual and source editors
- Create basic markdown elements
- Create and modify markdown templates for MS Word

Resources from PowerPoint

- A Brief History of R Markdown
- Pandoc documentation on type references
- Block and Inline Elements
- Command Line Essentials
- Polishing Documents
- email me for a copy: douglas.joubert@nih.gov

Configuration for Exercises

- Class features exercises that will help you learn by doing. Install the following on your machine:
  - Latest version of RStudio, v2022.07.0-548 or later
  - Latest version of Quarto (v1.0.36 or greater)
- Quarto is also available as a package. The **quarto** package provides an R interface to frequently used operations in the Quarto Command Line Interface (CLI).

```
install.packages("quarto")
```

Configuration: Additional Packages

- knitr, tool for dynamic report generation in R
- rmarkdown, helps you create dynamic analysis documents that combine code, rendered output (such as figures).
- tidyverse, includes the packages that you’re likely to use in everyday data analyses.

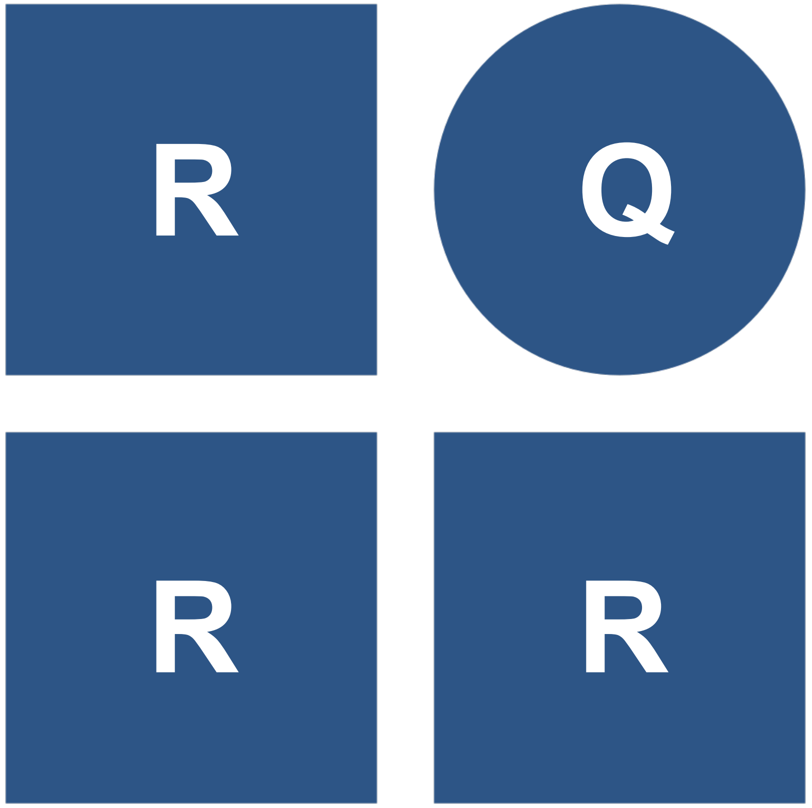
Literate Programming and Markdown

Markdown

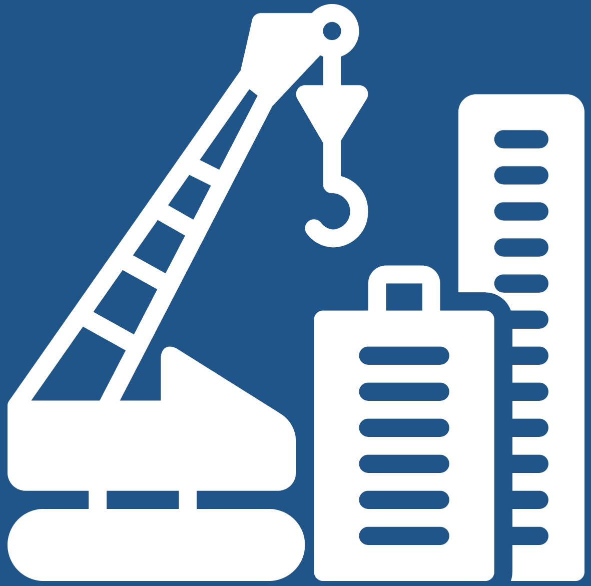
- R Markdown, a light-weight markup language for creating documents in multiple formats
- When a document is processed by knitr, chunks of code are executed, and graphs or other results will be inserted into the final document
- A process called “literate programming”

How Quarto is Different From R Markdown

- Quarto is compute-agnostic
- The ecosystem of R packages is replaced by a single framework
- Easier to organize appearance across documents
- Features (e.g. figures, tables) have better cross-format support
- Up-to-date revealjs slides
- Easier to customize websites and books with projects



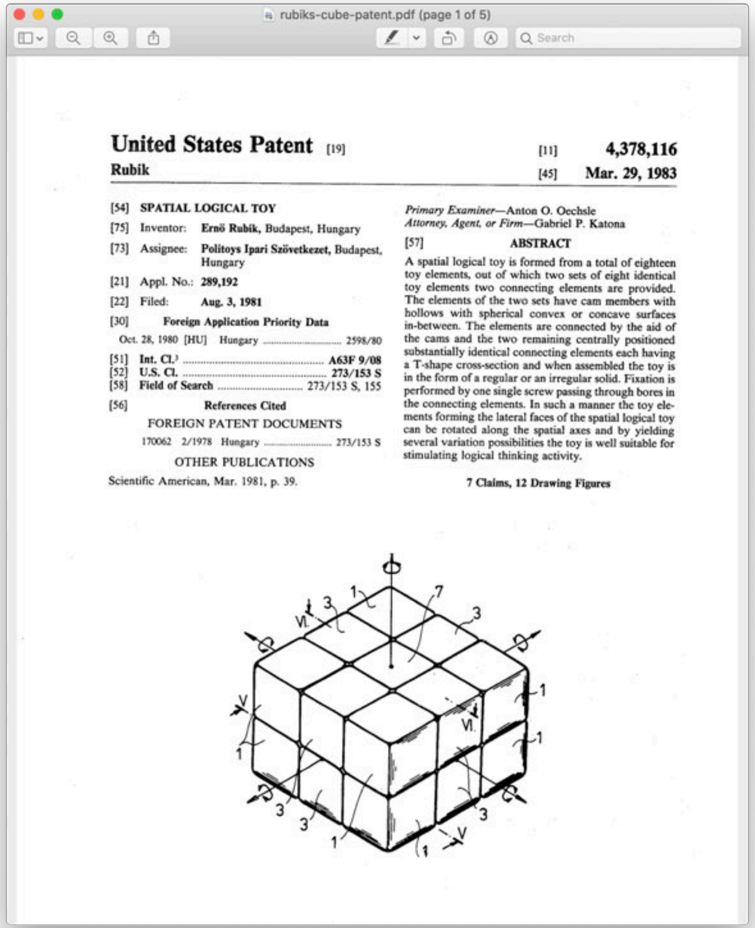
Creating Quarto Documents



Quarto Docs

- Quarto document (.qmd) is a plain text file, that can be rendered to many different formats
- Like R Markdown, Quarto uses Knitr to execute R code
- Can render most existing .rmd files without modification

Quarto Content Structure



United States Patent [19] [11] **4,378,116**
Rubik [45] **Mar. 29, 1983**

[54] **SPATIAL LOGICAL TOY**
[75] Inventor: **Ernö Rubik,** Budapest, Hungary
[73] Assignee: **Politoys Ipari Szövetkezet,** Budapest, Hungary
[21] Appl. No.: **289,192**
[22] Filed: **Aug. 3, 1981**
[30] **Foreign Application Priority Data**
Oct. 28, 1980 [HU] Hungary 2598/80
[51] **Int. Cl.³** **A63F 9/08**
[52] **U.S. Cl.** .... **273/153 S**
[58] **Field of Search** .... 273/153 S, 155
[56] **References Cited**
FOREIGN PATENT DOCUMENTS
170062 2/1978 Hungary .... 273/153 S
OTHER PUBLICATIONS
Scientific American, Mar. 1981, p. 39.

Primary Examiner—Anton O. Oechsle
Attorney, Agent, or Firm—Gabriel P. Katona

[57] **ABSTRACT**

A spatial logical toy is formed from a total of eighteen toy elements, out of which two sets of eight identical toy elements two connecting elements are provided. The elements of the two sets have cam members with hollows with spherical convex or concave surfaces in-between. The elements are connected by the aid of the cams and the two remaining centrally positioned substantially identical connecting elements each having a T-shape cross-section and when assembled the toy is in the form of a regular or an irregular solid. Fixation is performed by one single screw passing through bores in the connecting elements. In such a manner the toy elements forming the lateral faces of the spatial logical toy can be rotated along the spatial axes and by yielding several variation possibilities the toy is well suitable for stimulating logical thinking activity.

7 Claims, 12 Drawing Figures

Content: text, code, graphics

Structure: paragraphs, lists, emphasis, etc

Appearance: fonts, colors, layout

Format: based on function

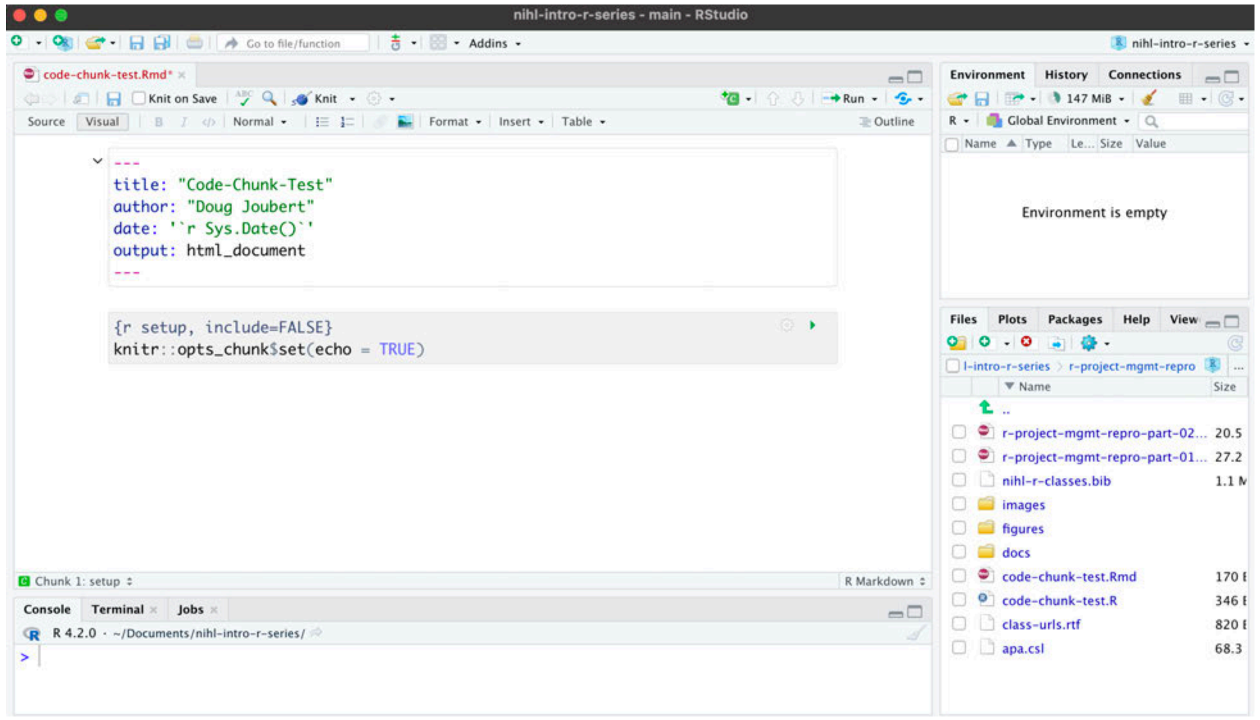
Layout (Advanced)

- Block
- Inline

Editor Options

Visual Editor

- Quarto visual editor provides a WYSIWYM editing interface
- The visual editor also includes support for executing code cells and viewing their output inline



```
---
title: "Code-Chunk-Test"
author: "Doug Joubert"
date: '`r Sys.Date()`'
output: html_document
---
```

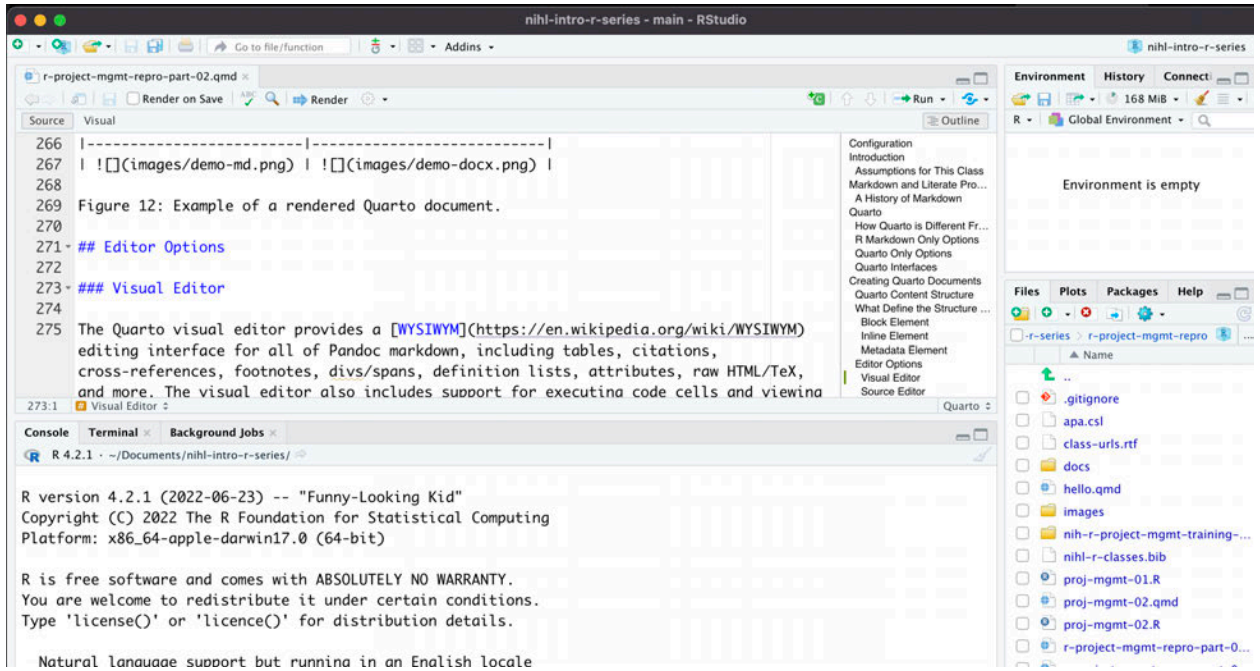
```
{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```
R 4.2.0 · ~/Documents/nihl-intro-r-series/
>
```

Name	Size
..	
r-project-mgmt-repro-part-02...	20.5
r-project-mgmt-repro-part-01...	27.2
nihl-r-classes.bib	1.1 M
images	
figures	
docs	
code-chunk-test.Rmd	170
code-chunk-test.R	346
class-urls.rtf	820
apa.csl	68.3

Source Editor

- Same document using the "source editor" mode
- The symbols scattered throughout the text are examples of markdown
- You should know how to use the source editor, since you might have to fix a piece of broken markdown



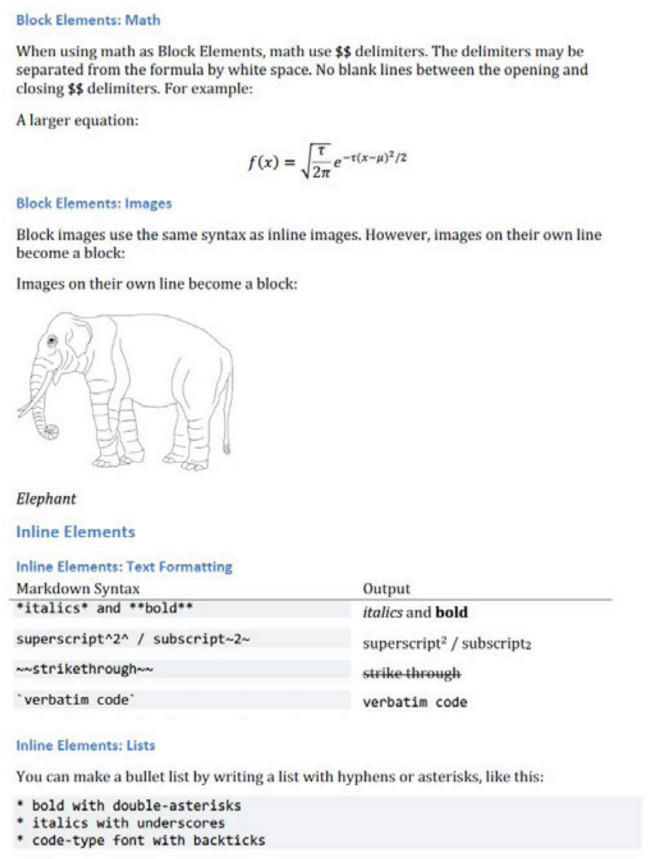
```
|-------------------------|--------------------------|
| ![](images/demo-md.png) | ![](images/demo-docx.png) |

Figure 12: Example of a rendered Quarto document.

## Editor Options

### Visual Editor

The Quarto visual editor provides a [WYSIWYM](https://en.wikipedia.org/wiki/WYSIWYM)
editing interface for all of Pandoc markdown, including tables, citations,
cross-references, footnotes, divs/spans, definition lists, attributes, raw HTML/TeX,
and more. The visual editor also includes support for executing code cells and viewing
```

```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale
```

Basic Markdown

Examples in PowerPoint

- I will show you some basic editing using the visual editor
- Example for each major Quarto element are including in student version of PowerPoint



Block Elements: Math

When using math as Block Elements, math use $$ delimiters. The delimiters may be separated from the formula by white space. No blank lines between the opening and closing $$ delimiters. For example:

A larger equation:

$$f(x) = \sqrt{\frac{\tau}{2\pi}} e^{-\tau(x-\mu)^2/2}$$

Block Elements: Images

Block images use the same syntax as inline images. However, images on their own line become a block:

Images on their own line become a block:

Elephant

Inline Elements

Inline Elements: Text Formatting

Markdown Syntax	Output
`*italics* and **bold**`	*italics* and **bold**
`superscript^2^ / subscript~2~`	superscript^2 / subscript$_2$
`~~strikethrough~~`	~~strike through~~
`` `verbatim code` ``	`verbatim code`

Inline Elements: Lists

You can make a bullet list by writing a list with hyphens or asterisks, like this:

```
* bold with double-asterisks
* italics with underscores
* code-type font with backticks
```

Metadata: YAML

- Metadata can be included via YAML
- Human-readable data-serialization language
- Commonly used for configuration files
- Example is the header of a Quarto document

```yaml
---
title: "US Patent: A Spatial Toy"
author:
  - Erno Rubrik
  - Albert Einstein
format:
  html:
    toc: true
    abstract: |
      This is the abstract.

      It has two paragraphs.
---
```

Metadata: YAML Syntax Rules

- Set single option key: value
- Strings with : must be quoted
- Include multiple values in a list with -
- Nest key-value pairs using indentation
- Two resources:
  - Pandoc interpretation of YAML metadata:
  - Overview of YAML syntax

Headers

You can make section headers of different sizes by initiating a line with some number of # symbols

```
# Title
    ## Main section
    ### Sub-section
    #### Sub-sub section
```

Title

Main section

Sub-section

Sub-sub section

Bullet: Lists

- Bullet list with hyphens or asterisks:

-
  ```
  * bold with double-asterisks
  * italics with underscores
  * code-type font with backticks
  ```
- or like this:

-
  ```
  - bold with double-asterisks
  - italics with underscores
  - code-type font with backticks
  ```

Each will appear as:

- bold with double-asterisks
- italics with underscores
- code-type font with backticks

be consistent. This maintains the readability of your code.

Lists: Numbered

Make a numbered list by just using numbers

```
1. bold with double-asterisks
2. italics with underscores
3. code-type font with
backticks
```

This will appear as:

1. bold with double-asterisks
2. italics with underscores
3. code-type font with backticks

Pandoc lists include *nesting*, *definition lists*, *chunks in lists*, and *example lists*. The Pandoc manual has detailed information about lists.

Text Formatting

Markdown Syntax	Output
italics and **bold**	italics and **bold**
superscript^2^ / subscript~2~	superscript^2 / subscript$_2$
~~strikethrough~~	strikethrough
`verbatim code`	verbatim code

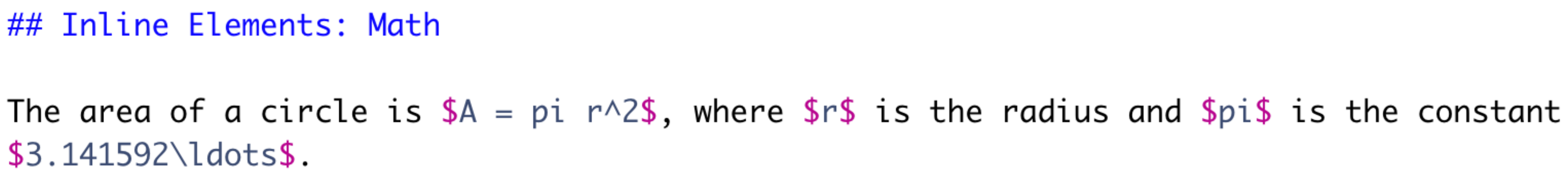
Math (Block Element)

Use `$$` delimiters. The delimiters may be separated from the formula by whitespace. No blank lines between the opening and closing `$$` delimiters.

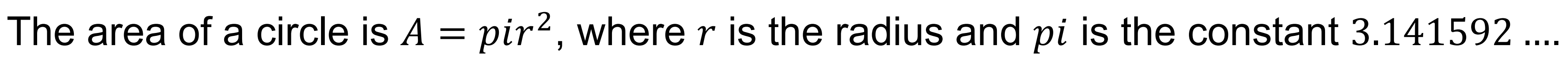
```
$$
f(x)={sqrt{frac{tau}{2pi}}}
  e^{-tau (x-mu )^{2}/2}
$$
```

$$f(x) = \sqrt{\frac{\tau}{2\pi}} e^{-\tau(x-\mu)^2/2}$$

Math: (Inline)



```
## Inline Elements: Math

The area of a circle is $A = pi r^2$, where $r$ is the radius and $pi$ is the constant
$3.141592\ldots$.
```



The area of a circle is $A = pi r^2$, where r is the radius and pi is the constant $3.141592\ldots$.

- TeX math occurs between two $
- Opening $ must have a non-space character immediately to its right
- Closing $ must have a non-space character immediately to its left, and must not be followed immediately by a digit.

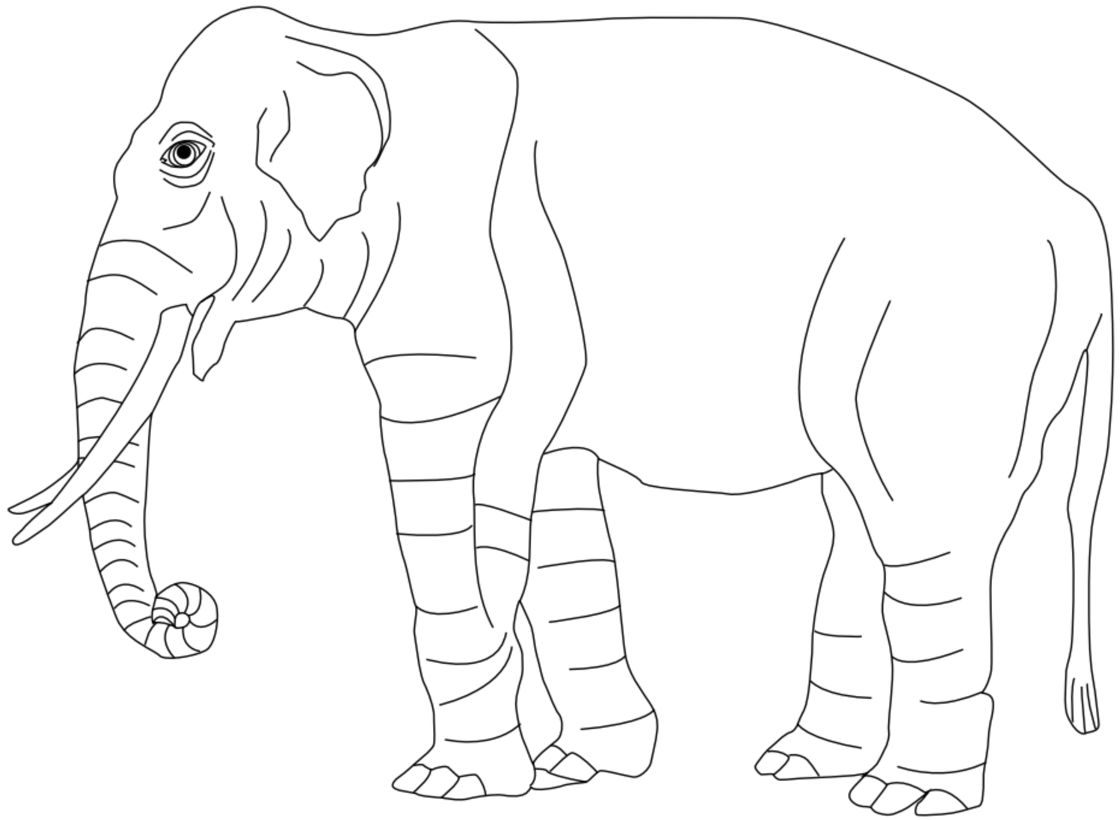
Links

- You can embed links with names [Links in Quarto](https://quarto.org/docs/reference/formats/html.html#links)

- You can also used direct links <https://quarto.org/docs/reference/formats/html.html#links>

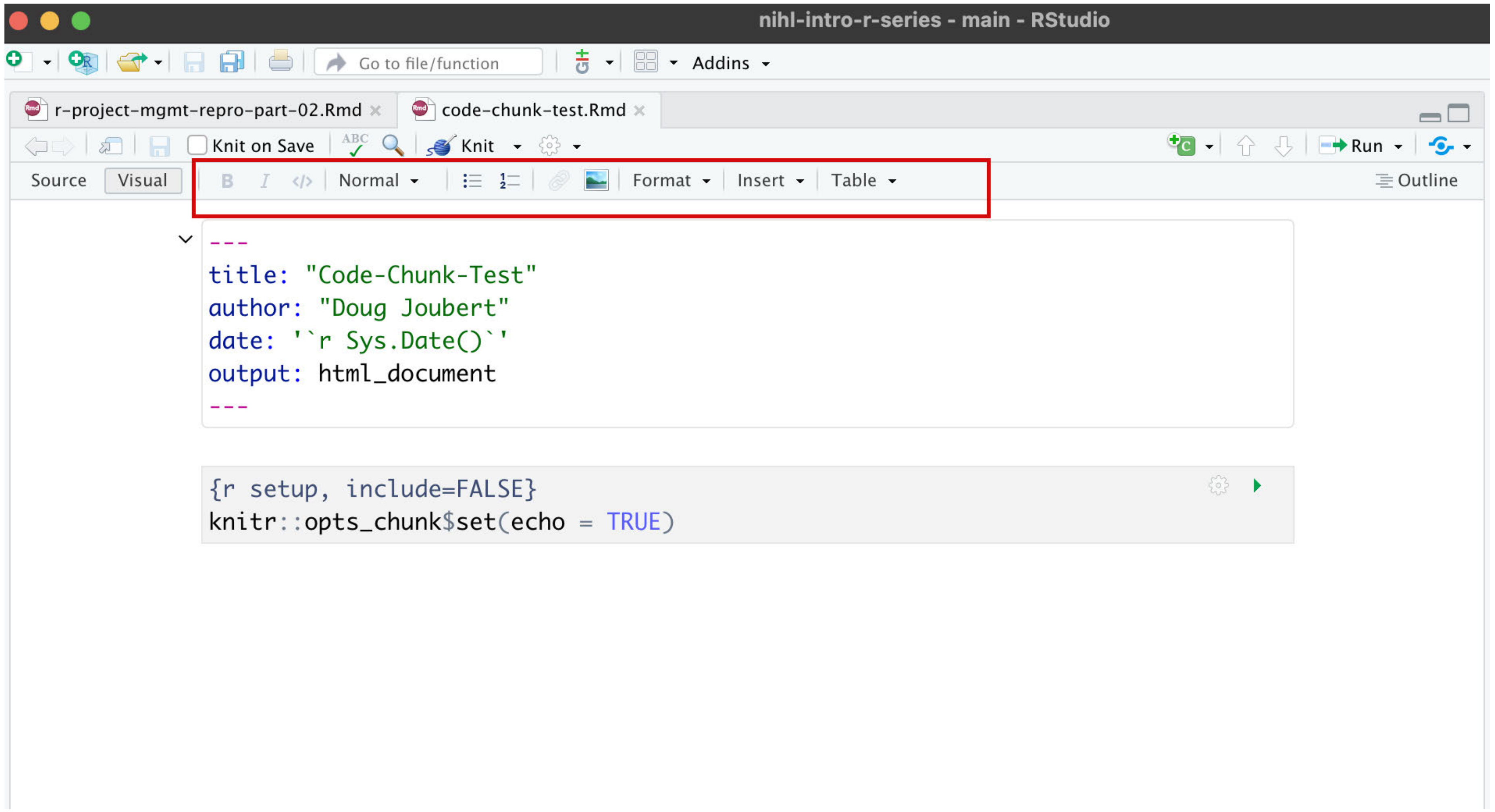
Images

- Like links, except that you start with a bang !
- The text within the [] provides a caption for the embedded image
- Images on their own line become a block



```
![Elephant](images/elephant.png){width="220"}
```

Lets Explore the Visual Editor in RStudio



```
---
title: "Code-Chunk-Test"
author: "Doug Joubert"
date: '`r Sys.Date()`'
output: html_document
---
```

```
{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Inserting Code Chunks Using R

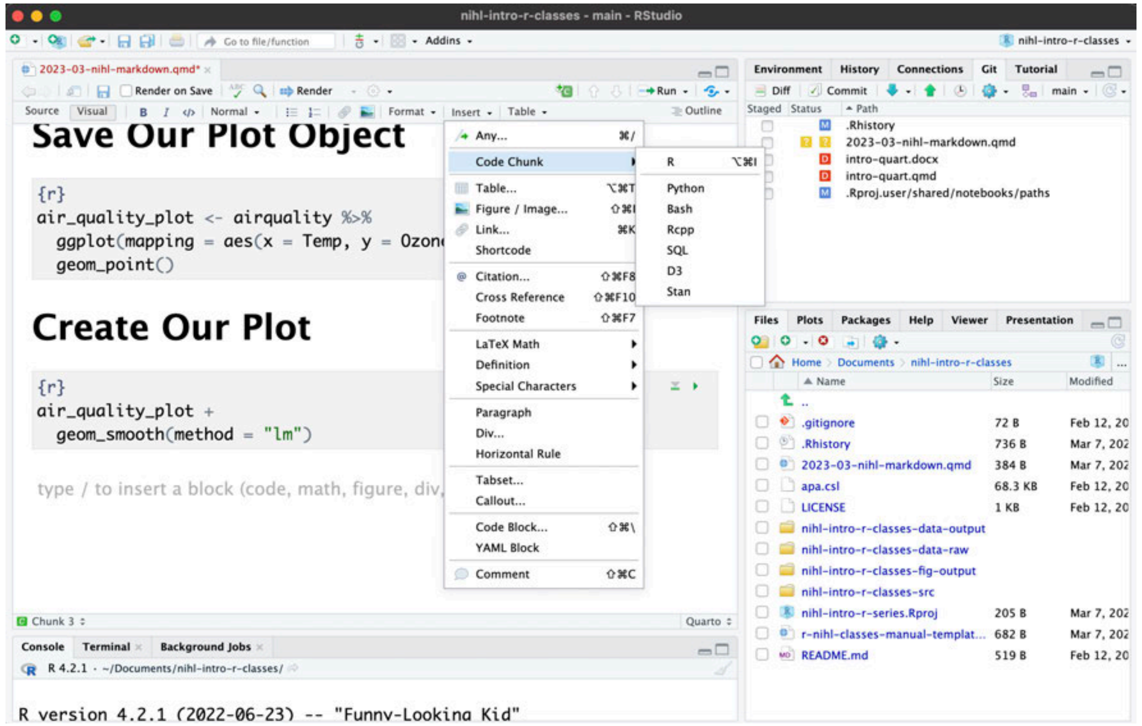
Inserting Code Chunks

- Insert chunks using keyboard shortcuts
  - Ctrl + Alt + I (Wintel)
  - Cmd + Option + I (iOS)
- Add Chunk command in the editor toolbar
- Typing the chunk delimiters

Inserting Code Chunks: Visual View

NIH Library
Office of Research Services
Serving the NIH Community

- Add Chunk command in the editor toolbar
- `Insert>Code Chunk>R`



Save Our Plot Object

```
{r}
air_quality_plot <- airquality %>%
  ggplot(mapping = aes(x = Temp, y = Ozon
  geom_point()
```

Create Our Plot

```
{r}
air_quality_plot +
  geom_smooth(method = "lm")
```

type / to insert a block (code, math, figure, div,

```
R 4.2.1 · ~/Documents/nihl-intro-r-classes/

R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
```

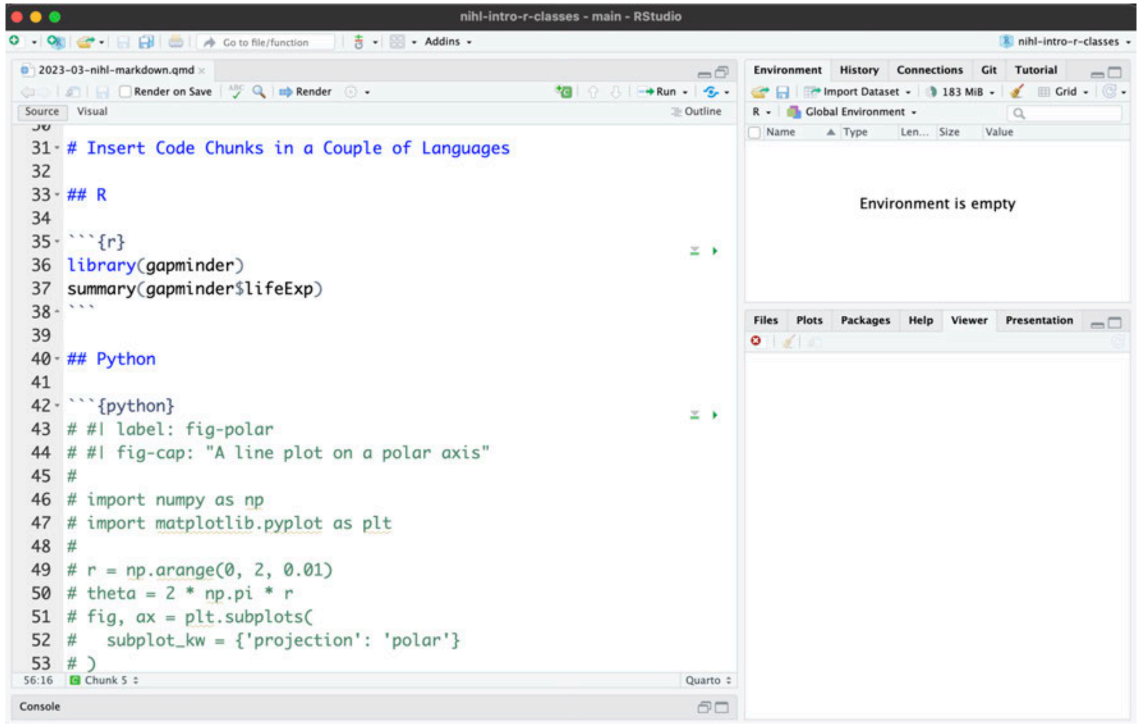
Git staged paths:
- .Rhistory
- 2023-03-nihl-markdown.qmd
- intro-quart.docx
- intro-quart.qmd
- .Rproj.user/shared/notebooks/paths

Name	Size	Modified
..		
.gitignore	72 B	Feb 12, 20
.Rhistory	736 B	Mar 7, 202
2023-03-nihl-markdown.qmd	384 B	Mar 7, 202
apa.csl	68.3 KB	Feb 12, 20
LICENSE	1 KB	Feb 12, 20
nihl-intro-r-classes-data-output		
nihl-intro-r-classes-data-raw		
nihl-intro-r-classes-fig-output		
nihl-intro-r-classes-src		
nihl-intro-r-series.Rproj	205 B	Mar 7, 202
r-nihl-classes-manual-templat...	682 B	Mar 7, 202
README.md	519 B	Feb 12, 20

Inserting Code Chunks: Source View

NIH Library
Office of Research Services
Serving the NIH Community

- Typing the chunk delimiters
- Adding language in the `{}`
- Example is showing both an `R` and a `Python` code chunk
- Make sure you add closing



~~~
# Insert Code Chunks in a Couple of Languages

## R

```{r}
library(gapminder)
summary(gapminder$lifeExp)
~~~

Python

```{python}
# #| label: fig-polar
# #| fig-cap: "A line plot on a polar axis"
#
# import numpy as np
# import matplotlib.pyplot as plt
#
# r = np.arange(0, 2, 0.01)
# theta = 2 * np.pi * r
# fig, ax = plt.subplots(
#   subplot_kw = {'projection': 'polar'}
# )
```

Code Chunk Output Options

Code Output Options: YAML

- Variety of options available for customizing output from executed code
- These options can be specified globally in the YAML header

```
---
title: "Example of Controlling Co
de Output in Header"
execute:
  echo: false
---
```

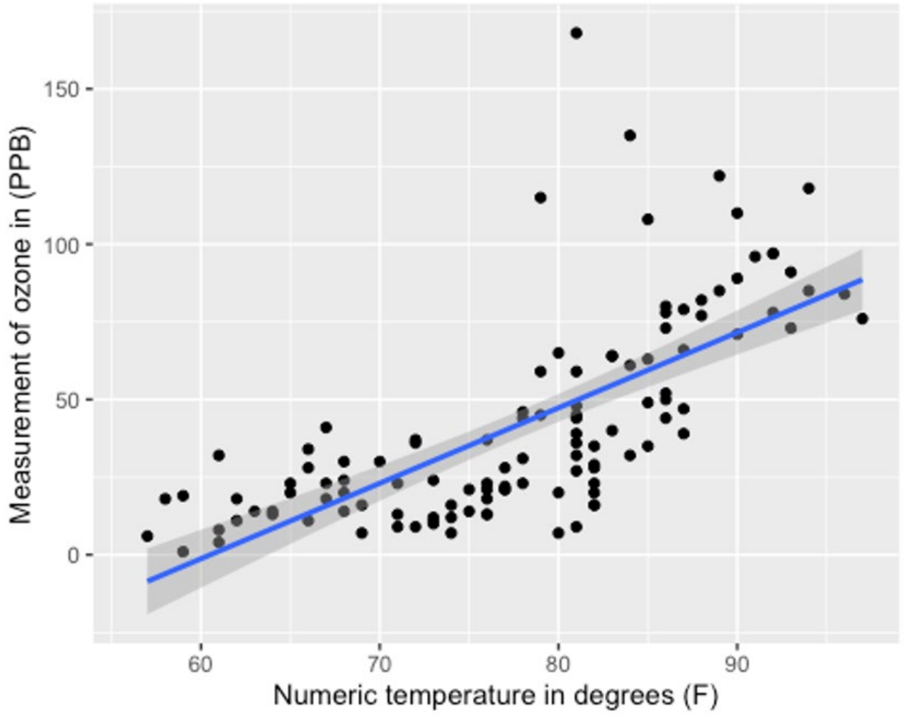
Code Output Options: Code Block

NIH Library
Office of Research Services
Serving the NIH Community

- Can override this YAML-level options on a per code-block basis
- Chunk options are included in a special comment at the top of the block, using #|

```
{r}
#| echo: true
air_quality_plot +
  geom_smooth(method = "lm")
```

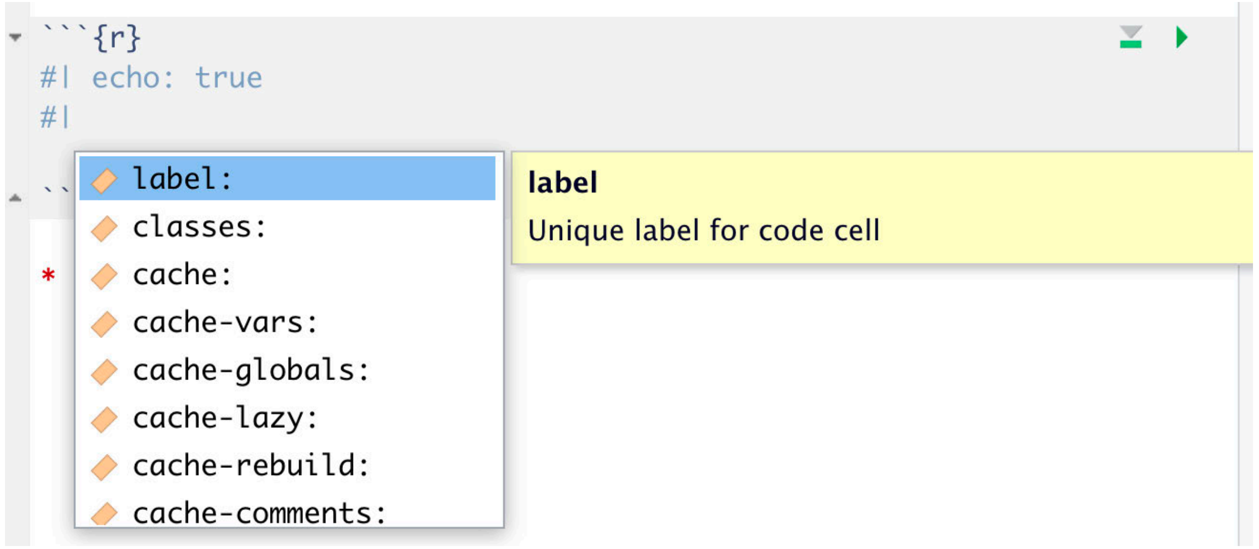
```
air_quality_plot +
  geom_smooth(method = "lm")
```



Code Output Options: Use Built-in Tools

NIH Library
Office of Research Services
Serving the NIH Community

- These hash pipe options are more portable across computational engines
- Don't forget the "help" option built into RStudio



~~~
```{r}
#| echo: true
#|
~~~

label
Unique label for code cell

Common Code Options

Option	Description
eval	Evaluate the code chunk (if false, just echos the code into the output).
echo	Include the source code in output
output	Include the results of executing the code in the output (`true`, `false`, or `asis` to indicate that the output is raw markdown
warning	Include warnings in the output.
error	Include errors in the output (errors that will not halt processing of the document)
include	Catch all for preventing any output (code or results) from being included (e.g. include: false suppresses all output from the code block).

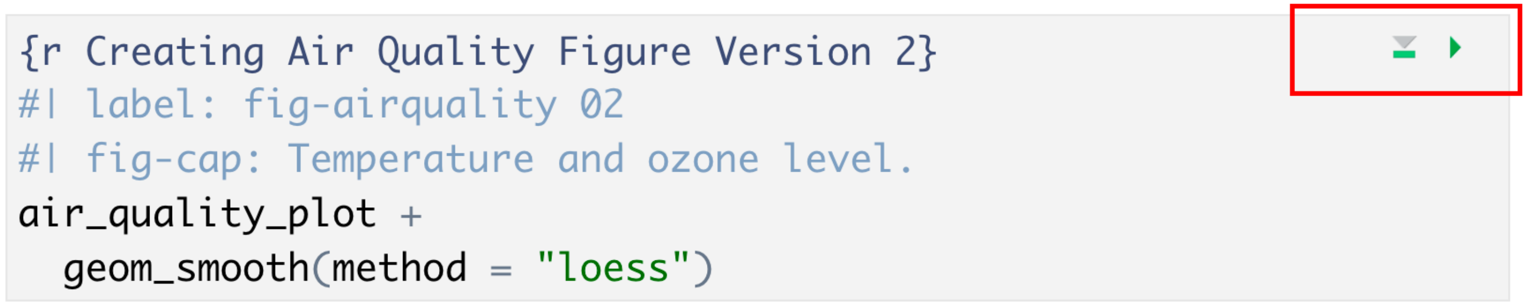
Running Code Chunks

3 Primary Options

- Run from code chunk play button
- Run Menu
- Keyboard shortcuts

Running Code Chunks: Play Button

- Run from code chunk (green play button on the right top corner)
- Allows us to run one specific code chunk

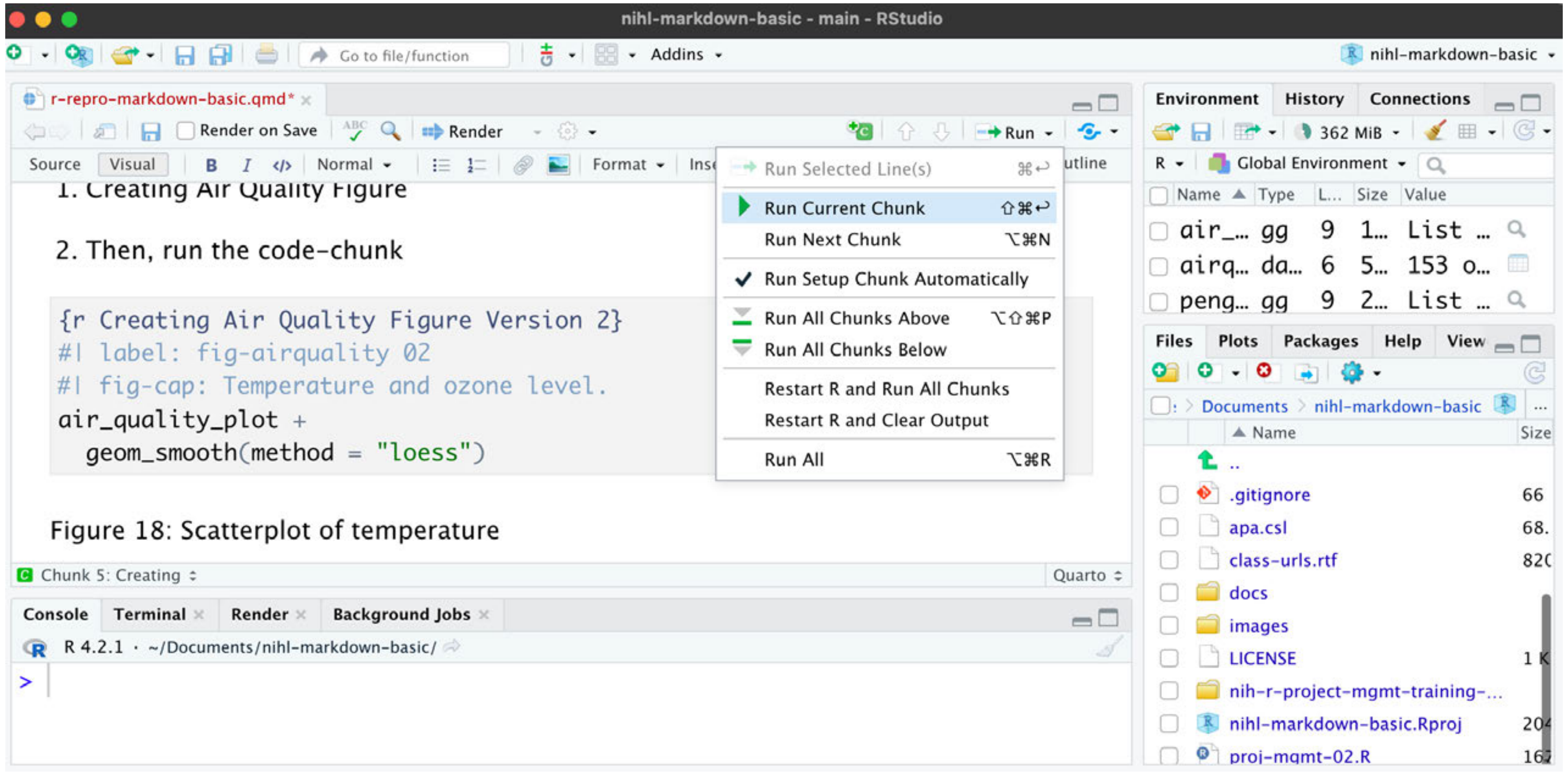


```
{r Creating Air Quality Figure Version 2}
#| label: fig-airquality 02
#| fig-cap: Temperature and ozone level.
air_quality_plot +
  geom_smooth(method = "loess")
```

Running Code Chunks: Run Menu

NIH Library
Office of Research Services
Serving the NIH Community

- Run code chunk from the Code-Chunk menu



1. Creating Air Quality Figure

2. Then, run the code-chunk

```
{r Creating Air Quality Figure Version 2}
#| label: fig-airquality 02
#| fig-cap: Temperature and ozone level.
air_quality_plot +
  geom_smooth(method = "loess")
```

Figure 18: Scatterplot of temperature

```
R 4.2.1 · ~/Documents/nihl-markdown-basic/
>
```

Name	Type	L...	Size	Value
air_...	gg	9	1...	List ...
airq...	da...	6	5...	153 o...
peng...	gg	9	2...	List ...

Name	Size
..	
.gitignore	66
apa.csl	68.
class-urls.rtf	82(
docs	
images	
LICENSE	1 K
nih-r-project-mgmt-training-...	
nihl-markdown-basic.Rproj	204
proj-mgmt-02.R	167

Running Code Chunks: Keyboard

Task	Windows & Linux	macOS
Create a code chunk	Ctrl + Alt + I	Cmd + Option + I
Run all chunks above	Ctrl+Alt+P	Command+Option+P
Run current chunk	Ctrl+Alt+C	Command+Option+C
Run current chunk	Ctrl+Shift+Enter	Command+Shift+Enter
Run next chunk	Ctrl+Alt+N	Command+Option+N
Run all chunks	Ctrl+Alt+R	Command+Option+R
Go to next chunk/title	Ctrl+PgDown	Command+PgDown
Go to previous chunk/title	Ctrl+PgUp	Command+PgUp

Figure Options

Figure Options

- Several ways to control the default width and height of figures
- Quarto sets a default width and height for figures appropriate to the target output format
- The table is displaying the defaults (expressed in inches)

Format	**Default**
Default	7 x 5
HTML Slides	9.5 x 6.5
HTML Slides (reveal.js)	9 x 5
PDF	5.5 x 3.5
PDF Slides (Beamer)	10 x 7
PowerPoint	7.5 x 5.5
MS Word, ODT, RTF	5 x 4
EPUB	5 x 4
Hugo	8 x 5

Figure Options: Header

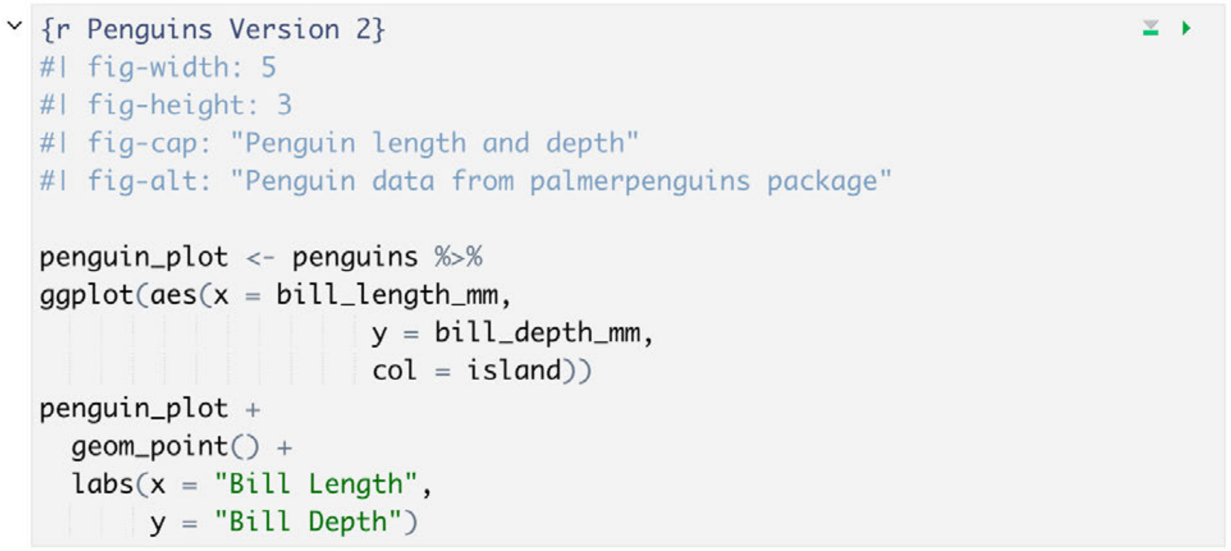
- Set the default sizes using the `fig-width` and `fig-height` options in YAML header

```
---
title: "My Document"
format:
  html:
    fig-width: 8
    fig-height: 6
  pdf:
    fig-width: 7
    fig-height: 5
---
```

Figure Options: Caption and Alt Text

NIH Library
Office of Research Services
Serving the NIH Community

- Specify the caption and alt text for figures generated from code using the `fig-cap` and `fig-alt` options



```
{r Penguins Version 2}
#| fig-width: 5
#| fig-height: 3
#| fig-cap: "Penguin length and depth"
#| fig-alt: "Penguin data from palmerpenguins package"

penguin_plot <- penguins %>%
ggplot(aes(x = bill_length_mm,
           y = bill_depth_mm,
           col = island))
penguin_plot +
  geom_point() +
  labs(x = "Bill Length",
       y = "Bill Depth")
```

Rendering

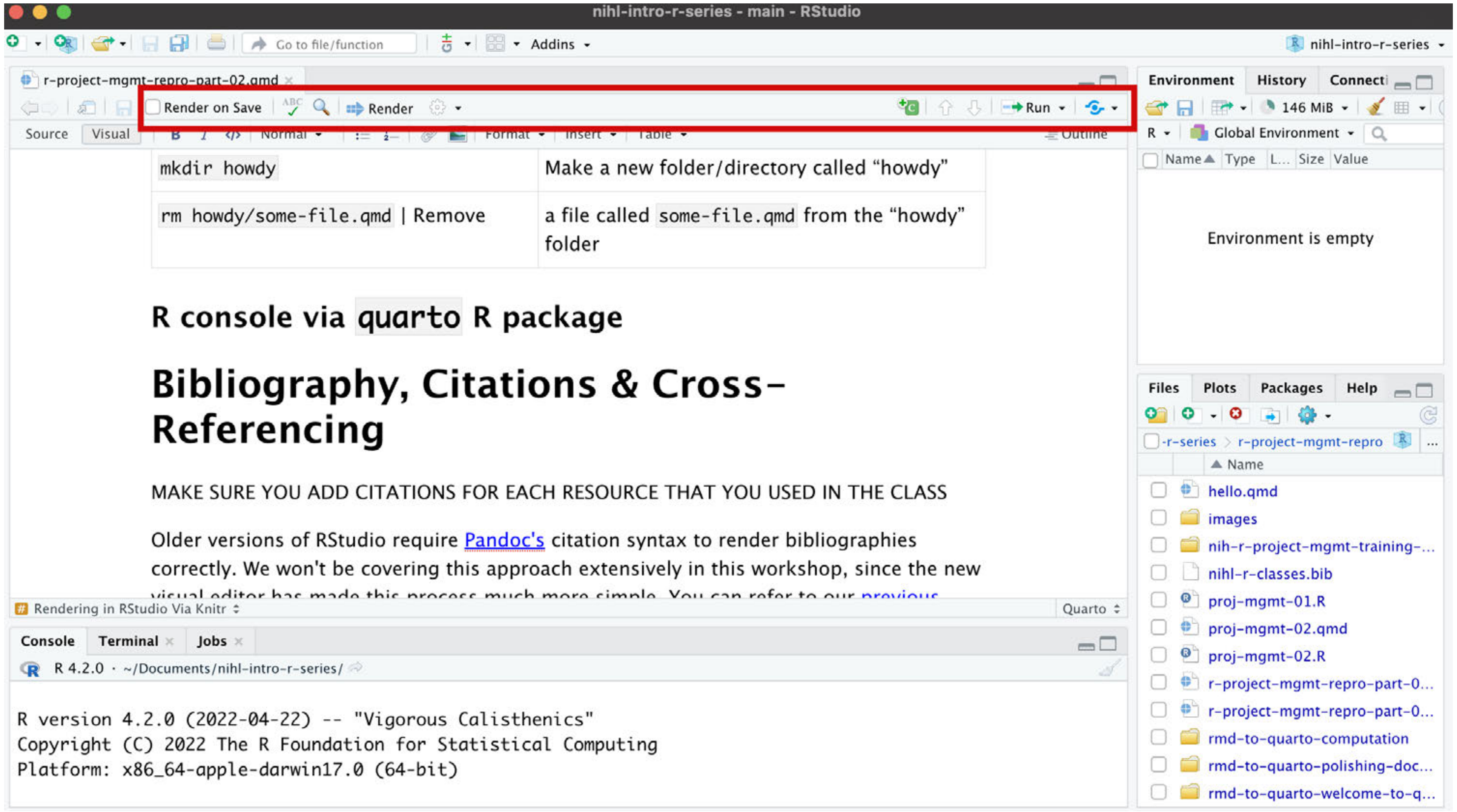
Rendering Options

- There are three methods for rendering a Quarto document.
  - Render in RStudio
  - System shell via quarto render
  - R console via quarto R package

Rendering in RStudio Using Knitr

- RStudio is integrated with R, knitr and Quarto
  - Render button
  - Visual Editor
  - Preview of output
- Quarto uses the Knitr engine just like R-markdown to execute R code natively
- Quarto can *also* use the Jupyter engine to natively execute Julia, Python, or other languages that Jupyter supports

Rendering in RStudio Options



`mkdir howdy`	Make a new folder/directory called "howdy"
`rm howdy/some-file.qmd` | Remove	a file called `some-file.qmd` from the "howdy" folder

R console via `quarto` R package

Bibliography, Citations & Cross-Referencing

MAKE SURE YOU ADD CITATIONS FOR EACH RESOURCE THAT YOU USED IN THE CLASS

Older versions of RStudio require Pandoc's citation syntax to render bibliographies correctly. We won't be covering this approach extensively in this workshop, since the new

Environment is empty

-r-series > r-project-mgmt-repro

- hello.qmd
- images
- nih-r-project-mgmt-training-...
- nihl-r-classes.bib
- proj-mgmt-01.R
- proj-mgmt-02.qmd
- proj-mgmt-02.R
- r-project-mgmt-repro-part-0...
- r-project-mgmt-repro-part-0...
- rmd-to-quarto-computation
- rmd-to-quarto-polishing-doc...
- rmd-to-quarto-welcome-to-q...

```
R 4.2.0 · ~/Documents/nihl-intro-r-series/

R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
```

Other Rendering Options

- Because Quarto is also a command line interface (CLI), you can also render via the terminal
  - PowerPoint provides some common commands
  - Overview of using the command line and navigating files/directories is available via the Data Carpentries
  - Interactive tool for understanding commands: https://tldr.ostera.io
- You can render from the R console by installing install the Quarto package

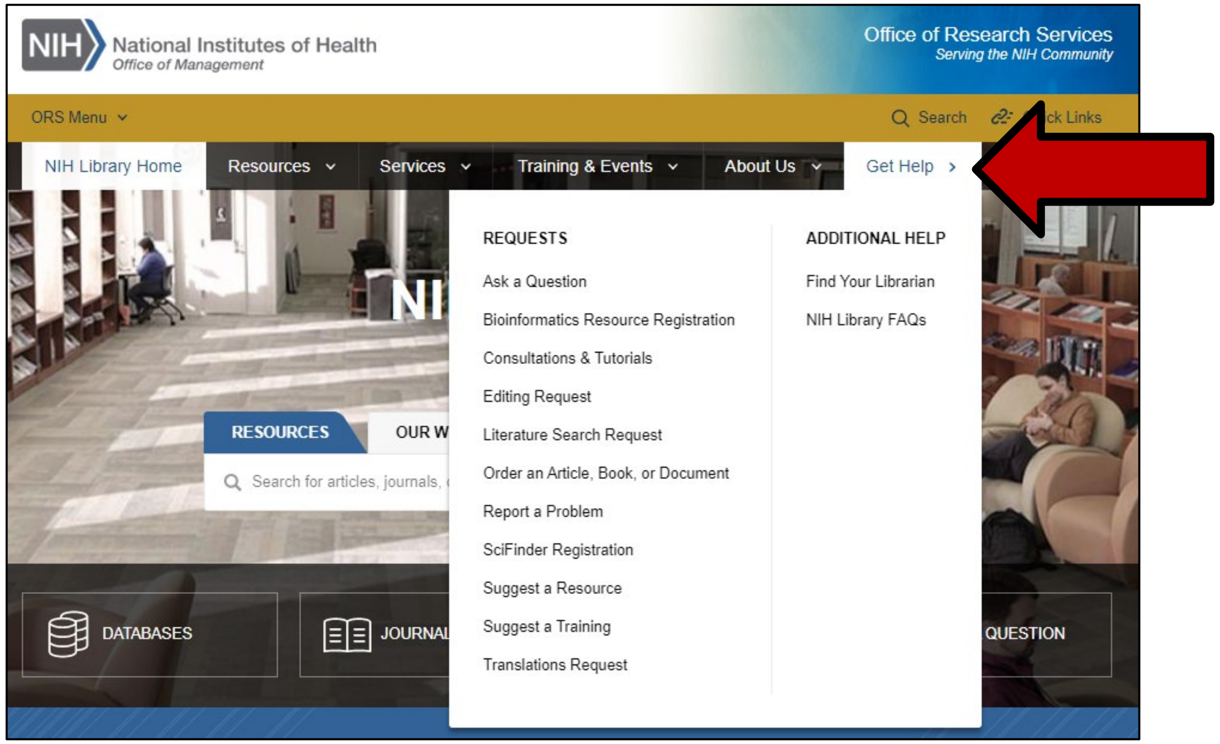
```
install.packages("quarto")
```

We Can Help

NIH Library
Office of Research Services
Serving the NIH Community

- Classes on a variety of data-related topics, including:
  - Data management
  - Data visualization
  - Data analysis
  - R and RStudio
- Computers which offers a suite of tools for data analysis, processing, and visualization

Contact Us for Ongoing Support



REQUESTS
- Ask a Question
- Bioinformatics Resource Registration
- Consultations & Tutorials
- Editing Request
- Literature Search Request
- Order an Article, Book, or Document
- Report a Problem
- SciFinder Registration
- Suggest a Resource
- Suggest a Training
- Translations Request

ADDITIONAL HELP
- Find Your Librarian
- NIH Library FAQs

Doug Joubert
Bioinformatics Support Program
301-827-3829
douglas.joubert@nih.gov

NIH Library Help Desk
(301) 496-1080

- **Ask a Question:** https://www.nihlibrary.nih.gov/get-help/ask-question
- **Request a Tutorial:** https://www.nihlibrary.nih.gov/get-help/consultations-tutorials
- **Sign up for Additional Classes:** https://www.nihlibrary.nih.gov/training/calendar

References and Further Reading

- A Brief History of R Markdown: a presentation by Yihui Xie in 2021.
- Pandoc documentation on type references: includes a compendium of the different block and inline elements recognized by Pandoc.
- Block and Inline Elements: provides clear description of elements in the setting of HTML.
- Command Line Essentials: a short primer on how to navigate through a file structure at the command line
- The Polishing Documents section of the RMD to Quarto workshop has some great information. However, I think it would work best as an advanced class.

Questions & Comments

DEPARTMENT OF HEALTH & HUMAN SERVICES · USA

NIH National Institutes of Health
Office of Management