

Intro to Quarto

Welcome!

Housekeeping!

- Be kind and curious!
- Slack and Zoom chat
- Ask questions

Schedule (Day 1)

Time	Activity
9:00 - 10:30	Intro to Quarto & brand.yml
10:30 - 11:00	<i>Coffee break</i> 
11:00 - 12:30	Presentations & Dashboards
12:30 - 1:30	<i>Lunch break</i> 
1:30 - 3:00	PDFs & Websites
3:00 - 3:30	<i>Coffee break</i> 
3:30 - 5:00	Customizing your brand

Schedule (Day 2)

Time	Activity
9:00 - 10:30	Intro to Quarto & brand.yml
10:30 - 11:00	<i>Coffee break</i> 
11:00 - 12:30	Presentations & Dashboards
12:30 - 1:30	<i>Lunch break</i> 
1:30 - 3:00	PDFs & Websites
3:00 - 3:30	<i>Coffee break</i> 
3:30 - 5:00	Customizing your brand

About me

Andrew Heiss

andrewheiss.com [@andrew.heiss.phd](https://www.linkedin.com/in/andrewheiss)

[@andrewheiss](https://twitter.com/andrewheiss) [andrewheiss](https://www.linkedin.com/in/andrewheiss)

- Assistant professor of public policy, Georgia State University
- Data visualization, statistics, and causal inference



Meeting you where you are

This course is designed for someone who:

- Knows some R or Python
- Maybe has an idea for a website
- Is relatively new to Quarto
- Wants to customize Quarto output

You'll learn:

- What Quarto is and how to use it
- How to create and publish websites with Quarto
- How to customize Quarto output

Course structure

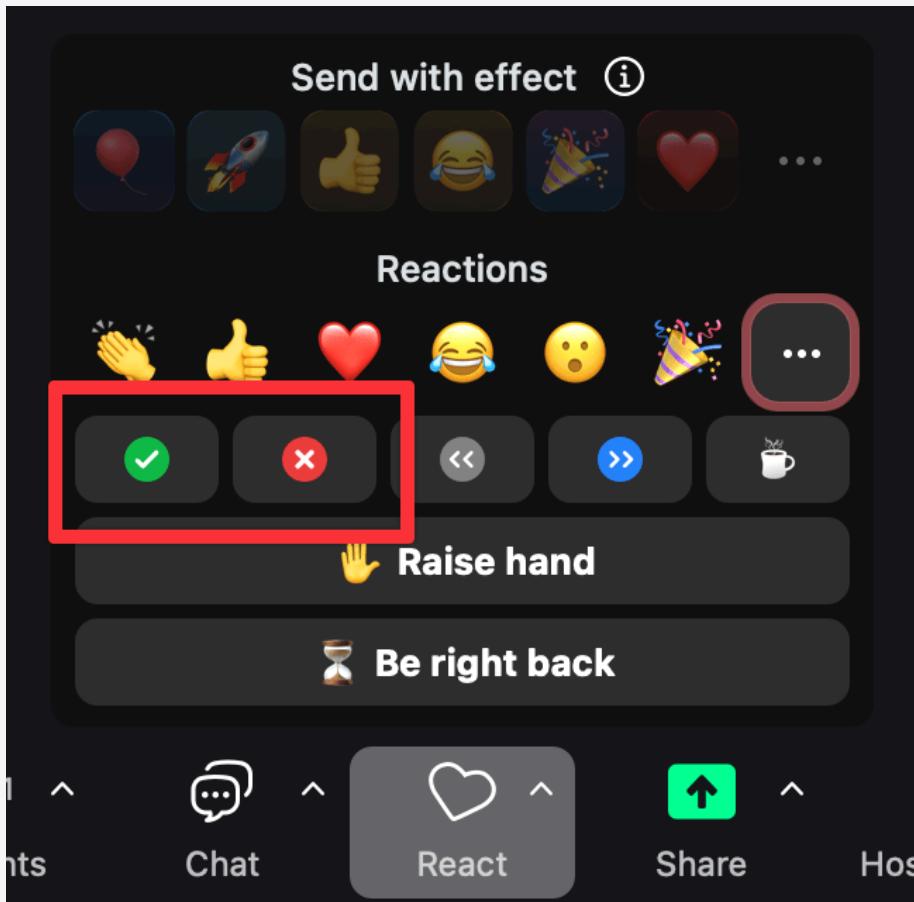
My turn

- Lecture segments
- Feel free to just watch, take notes, browse docs, or tinker around with the code

Your turn

- Exercises for you to do
- Work on your own or with others

Getting help



Use Zoom reactions

- =
"I'm stuck and need help!"
- =
"I finished the exercise"

Ask longer, more detailed questions in Slack

Your turn

Introduce yourself:

- Name
- Professional affiliation
- On a scale of 1-10, how well do you know...
 - Quarto?
 - R?
 - Python?
 - HTML and CSS?
- What do you hope to get out of this course?

04 : 00

Introduction to Quarto

Quarto is an...
open-source
scientific and technical
publishing system
built on Pandoc.

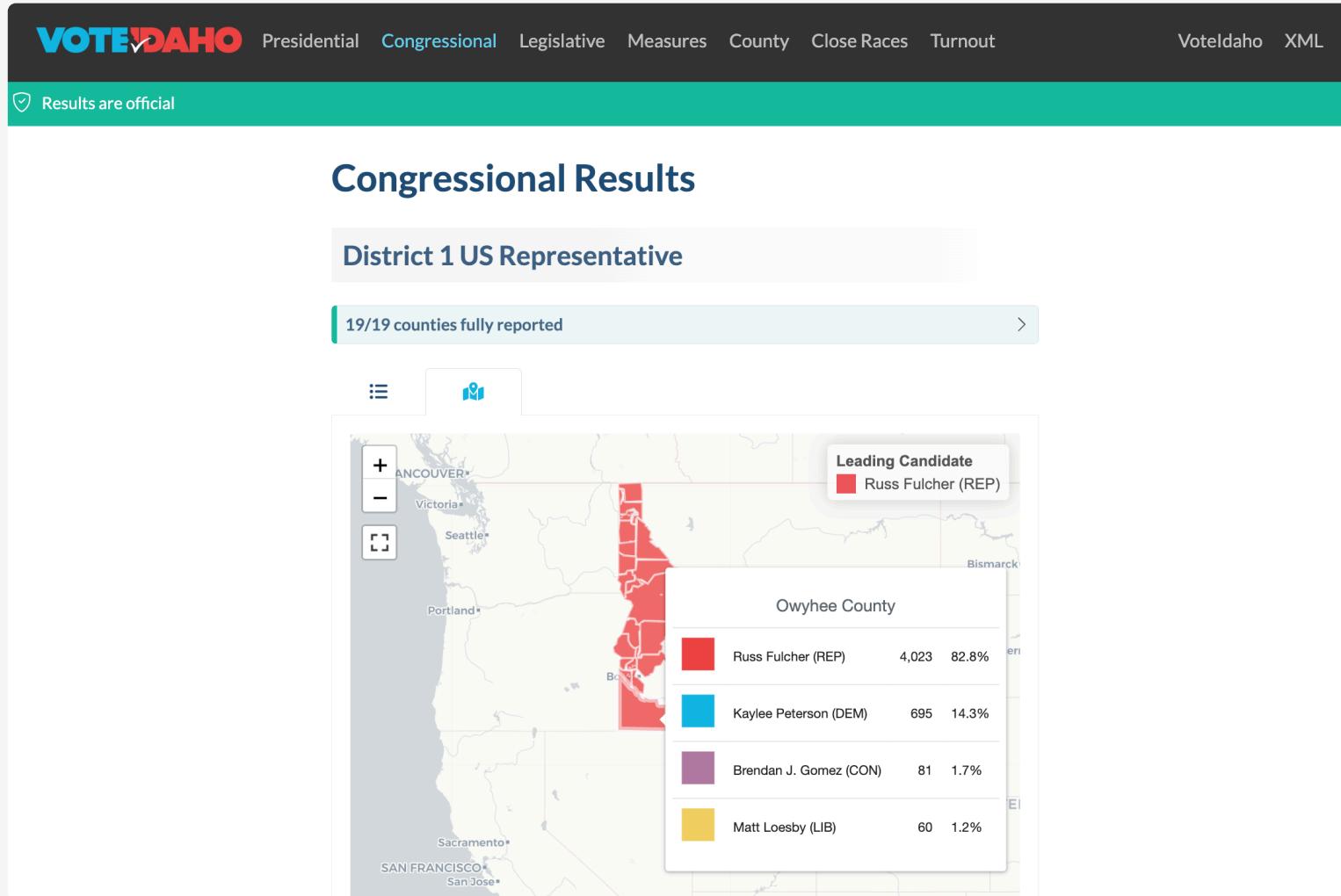
With Quarto...

...you can **weave** together **narrative** and **code** to produce elegantly formatted output such as documents, web pages, blog posts, books, dashboards, and more.

- Create dynamic content with Python, R, Julia, and Observable
- Edit documents in your favorite editor
- Publish technical content in HTML, PDF, Microsoft Word, and more
- Share content by publishing to the internet

Display data and results

2024 Idaho election results, by Gabe Osterhout and Andrew Heiss



Walk through a story

Council Housing & Neighborhood Income Inequality in Vienna by Tamara Premrov and Matthias Schnetzer (European Centre for Social Welfare Policy and Research, Austria)



Publish papers

Navigating Hostility: The Effect of Nonprofit Transparency and Accountability on Donor Preferences in the Face of Shrinking Civic Space

by Suparna Chaudhry, Marc Dotson, and Andrew Heiss

[HTML](#)[PDF](#)[Word](#)

Modeling and estimands

We analyze the results using a multilevel Bayesian multinomial model (see the appendix for complete model details). Our experimental data has a natural hierarchical structure, with 3 questions nested inside 12 separate experimental tasks, nested inside each of the 1,016 respondents, which lends itself to multilevel modeling (Jensen et al., 2021). Since it was impossible for every respondent to see every possible choice, we limit our analysis to the 576 combinations of feature levels (2 transparency × 2 accountability × 3 government relationships × 4 organizations × 4 issues × 3 funding) and use the multinomial model to calculate predicted probabilities of choice selection for each combination of possible treatment values. We then collapse this set of predicted probabilities into estimated marginal means (EMMs) for specific features of interest while marginalizing or averaging over all other predicted variables (Arel-Bundock et al., 2024; Leeper et al., 2020). This marginalization process allows us to isolate the statistical effect of each feature in isolation. We include a complete table of model results in Table A5, along with a brief illustration of converting from regression

Table of contents
Introduction
What determines individual donor behavior?
Research design
Sample
Experimental design
Modeling and estimands
Results
Discussion
Conclusion
Statements and declarations
References

Other formats
PDF (hikmat)
Manuscript PDF (hikmat-manuscript)

Multinomial probability of selection of choice, in respondent
Choice_{i,j} ~ Categorical({μ_{1,i,j}, μ_{2,i,j}, μ_{3,i,j}})

Model for probability of each option
{μ_{1,i,j}, μ_{2,i,j}, μ_{3,i,j}} = (β₀ + b_{0j}) + β_{1,2,3}Organization_{i,j} + β_{4,5,6}Issue area_{i,j} + β₇Transparency_{i,j} + β₈Accountability_{i,j} + β_{9,10}Funding source_{i,j} + β_{11,12}Government relationship_{i,j}

b_{0j} ~ N(0, σ₀) Respondent-specific offsets from global probability

Priors
β_{0...12} ~ N(0, 3) Prior for choice-level intercept and coefficients
σ₀ ~ Exponential(1) Prior for between-respondent variability

We do not include any respondent-level covariates beyond the treatment variables. Because this is an experimental design, any statistical confounding is accounted for during the process of randomization and covariates should have no systematic effect on treatment effects. We do not work with the raw results of the multinomial model directly. Given the conjoint design, we instead create a complete balanced grid of all 576 combinations of feature levels (2 transparency × 2 accountability × 3 government relationships × 4 organizations × 4 issues × 3 funding) and use the model to calculate predicted probabilities of choice selection for each combination of possible treatment values. We then collapse this set of predicted probabilities into estimated marginal means (EMMs) for specific features of interest while marginalizing or averaging over all other predicted variables (Arel-Bundock et al., 2024; Leeper et al., 2020). This marginalization process allows us to isolate the statistical effect of each feature in isolation. We include a complete table of model results in Table A5, along with a brief illustration of converting from regression

Share research

Pandemic Pass? Treaty Derogations and Human Rights Practices During COVID-19 by Suparna Chaudhry, Audrey Comstock, and Andrew Heiss

The screenshot shows a research project page titled "Pandemic Pass? Treaty Derogations and Human Rights Practices During COVID-19" by Suparna Chaudhry, Audrey Comstock, and Andrew Heiss. The page has a dark blue header with navigation links: "Pandemic Pass?", "Data and replication", "Analysis", "Presentations", "Paper", and a search bar. A sidebar on the left contains links for "Data", "Replication", and "Other details". The main content area is titled "Download final data" and describes three data formats: CSV, RDS, and DTA. It includes download links for "Weekly data" and "Quarterly data". At the bottom right, there is a link to "Data details".

Pandemic Pass? Data and replication Analysis Presentations Paper ▾

Data > Download final data

Download final data

The cleaned data is available in three formats:

- CSV file for any program
- `.rds` file for R (load with `df <- readRDS("weekly_panel.rds")`)
- `.dta` file for Stata (load with `use "weekly_panel.dta"`)

Weekly data

[weekly_panel.csv](#) [weekly_panel.rds](#) [weekly_panel.dta](#)

Quarterly data

[quarterly_panel.csv](#) [quarterly_panel.rds](#) [quarterly_panel.dta](#)

Data details →

Teach classes

Data Science for the Social Sciences, Gov 50, Harvard University, taught by Matt Blackwell

Gov 50 Syllabus Schedule Staff Materials Assignments Resources Ed Gradescope Q

Data Science for the Social Sciences

Learning to use data to explore the social, political, and economic world

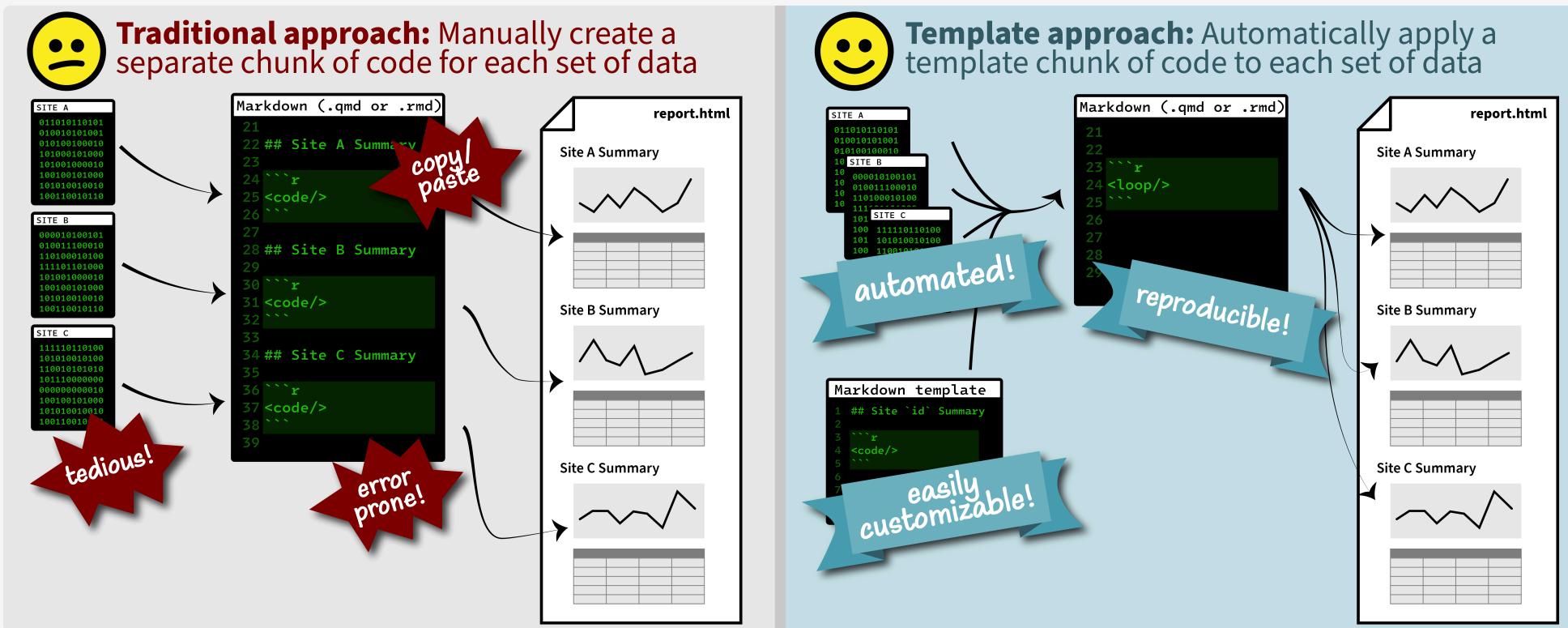
Gov 50 • Fall 2023
Harvard University



Instructor  Prof. Matt Blackwell  CGIS Knafel 305  mblackwell@gov.harvard.edu  matt_blackwell  Schedule an appointment	Course details  Tue/Thu  September 3rd-December 20th, 2023  12:00-1:15 PM  Emerson 105  Slack  Course Calendar	Contacting me General questions about the course should be posted to either the course Ed Discussion board or the course Slack. Someone on the teaching staff will attempt to respond to these messages within 25 hours, but also remember that life can be busy and chaotic for everyone (including me!), so if I don't respond right away, don't worry! For other issues (absences, etc),
---	---	---

Automate and reproduce your output

Duplicating Quarto elements with code templates to reduce copy and paste errors by Althea A. Archer (United States Geological Survey)



Get your team on the same page

We Converted Our Documentation to Quarto by Melissa Van Bussel (Statistics Canada)

We Converted our Documentation to Quarto - posit::conf(2023)



- Website for R/Python User Group
- Training resources
- Presentations

Gapminder Report

Health and Wealth Around the World

Gapminder data, 1952–2007

Introduction

This analysis shows trends in life expectancy and GDP per capita for 142 countries from XXXX to YYYY. The data was originally collected by Hans Rosling and the Gapminder foundation.

Continent-level trends

Average life expectancy increased substantially between XXXX and YYYY. Asia saw the biggest average increase (see [Table 1](#)).

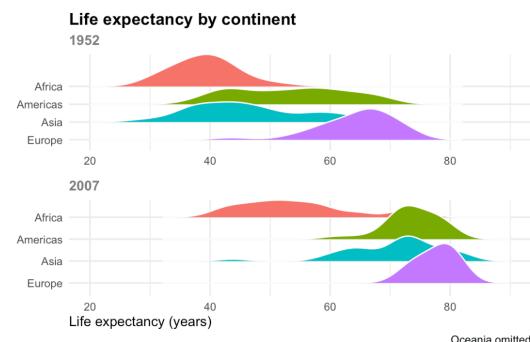


Figure 1: Distribution of life expectancy within continents

Table 1: Change in average life expectancy by continent

Continent	Averages		
	1952	2007	Change
Asia	46.3	70.7	24.4
Americas	53.3	73.6	20.3
Africa	39.1	54.8	15.7
Europe	64.4	77.6	13.2

This is what you'll work on today!

“Literate programming”

The screenshot shows the Quarto IDE interface with the following details:

- File Explorer:** Shows files in the project: `01-exercise.qmd`, `Data: gapminder`, `01-exercise.html`, and `01-exercise.qmd`.
- Editor:** The main editor pane displays the `01-exercise.qmd` file content. The code uses R Markdown syntax to generate a report titled "Health and Wealth Around the World" from Gapminder data (1952-2007). It includes sections for `format`, `html`, `knitr`, and `execute`. The `execute` section contains R code for loading libraries (`tidyverse`, `patchwork`, `ggridges`, `tinytable`) and running the analysis.
- Output:** The right side of the interface shows the generated report. It includes a title "Health and Wealth Around the World" and a subtitle "Gapminder data, 1952-2007". A note states: "This analysis shows trends in life expectancy and GDP per capita for `{n_countries}` countries from XXXX to YYYY. The data was originally collected by Hans Rosling and the Gapminder foundation." Below this is a section titled "Continent-level trends" with a note about average life expectancy increasing between XXXX and YYYY, mentioning Asia's biggest increase. It features two density heatmaps titled "Life expectancy by continent" for 1952 and 2007, showing the distribution of life expectancy within continents. A caption for Figure 1 reads: "Figure 1: Distribution of life expectancy within continents".
- Table:** A table titled "Averages" shows the average life expectancy and change for each continent in 1952 and 2007.
- Figure:** A scatter plot titled "Health and wealth in 2007" shows Life expectancy (years) on the Y-axis versus GDP per capita on the X-axis. Data points are colored by continent (Africa, Americas, Asia, Europe) and sized by country population. A regression line is shown. A legend indicates point sizes for populations of 250,000, 500,000, 750,000, 1,000,000, and 1,250,000.
- Bottom Status Bar:** Shows the current file is `main*`, line count is 40, character count is 31, and the status is "Open in Typora" and "Quarto: 1.8.20".

Why Quarto?

- Multilingual and independent of computational systems
- Quarto comes “**batteries included**” straight out of the box
- Consistent expression for core features
- Extension system
- Enable “single-source publishing”—create Word, PDFs, HTML, etc. from one source
- Use defaults that meet accessibility guidelines

Quarto formats

Feature	Quarto
Basic formats	html , pdf , docx , typst
Beamer	beamer
PowerPoint	pptx
HTML slides	revealjs
Advanced layout	Quarto Article Layout
Cross references	Quarto Crossrefs
Websites & blogs	Quarto Websites , Quarto Blogs
Books	Quarto Books
Interactivity	Quarto Interactive Documents
Journal articles	Journal Articles
Dashboards	Quarto Dashboards

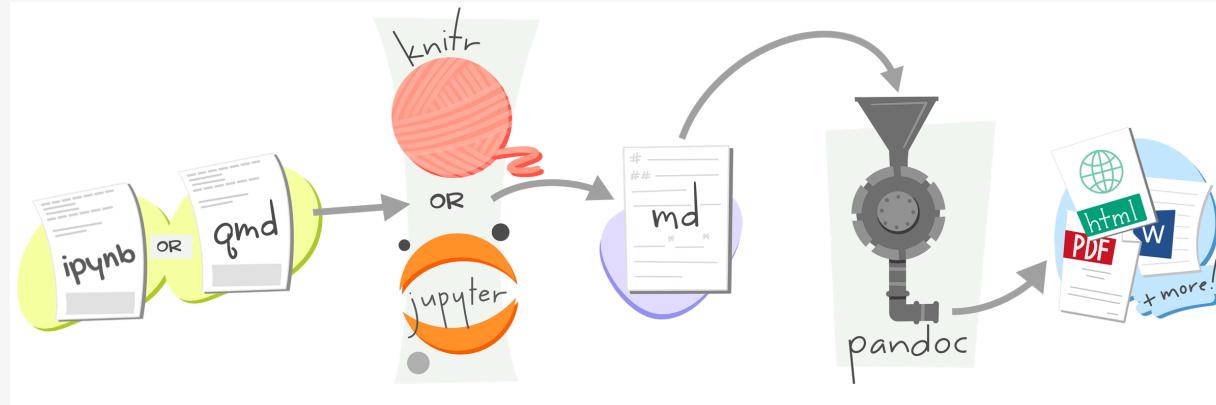
How it works

Quarto is a command line interface (**CLI**) that renders plain text formats ([.qmd](#), [.rmd](#), [.md](#)) OR mixed formats ([.ipynb](#)/Jupyter notebook) into static PDF/Word/HTML reports, books, websites, presentations and more.

```
1  Usage: quarto
2  Version: 1.8.24
3
4  Description:
5
6  Quarto CLI
7
8  Options:
9
10 -h, --help      - Show this help.
11 -V, --version   - Show the version number for this program.
12
13 Commands:
14
15 render          [input] [args...]
16 preview         [file] [args...]
17 serve           [input]
18 create          [type] [commands...]
19 create-project  [dir]
20 convert          <input>
21 pandoc          [args...]
22 typst           [args...]
23 run              [script] [args...]
24 add              <extension>
25 install          [target...]
```

- Render files or projects to various document types.
- Render and preview a document or website project.
- Serve a Shiny interactive document.
- **Create a Quarto project or extension**
- Create a project for rendering multiple documents
- Convert documents to alternate representations.
- Run the version of Pandoc embedded within Quarto.
- Run the version of Typst embedded within Quarto.
- Run a TypeScript, R, Python, or Lua script.
- Add an extension to this folder or project
- Installs an extension or global dependency.

Under the hood



- `jupyter` or `knitr` evaluates Python, Julia, R, or Observable code and returns a `.md` file along with the evaluated code
- Quarto applies Lua filters + CSS/LaTeX which is then evaluated alongside the `.md` file by Pandoc and converted to a final output format

Environment options

You have a couple options for following along today:

- 1. Posit Cloud** (RStudio in the cloud)
- 2. Local installation** (RStudio, Positron, or VS Code on your computer)

Your turn

Go to the course website and click on **Setup** in the sidebar.

andhs.co/quarto-websites-2025

Follow the instructions for either Option 1 or Option 2.

05 : 00

Quarto workflow

- Open a `.qmd` file.
- Preview/render the document.
- Make a change and preview/render again.



Render/preview

RStudio



⌘ ⌘ K



Ctrl + Shift + K

Positron / VS Code

Terminal



Your turn

- Open `01-exercise.qmd`
- Preview/render the document
 - If you're using Posit Cloud, you might be asked to allow pop-ups
- Edit the title and preview the document again.

05 : 00

Quarto documents

Anatomy of a Quarto document

.qmd file format with three components:

1. **YAML**: Metadata
2. **Text**: Markdown
3. **Code**: R, Python, Observable, and Julia

Weave it all together, and you have beautiful, powerful, and useful outputs!

Anatomy of a Quarto document

Metadata: YAML

```
my-document.qmd
```

```
1 ---
2 title: "My Cool Document"
3 format: html
4 ---
```

- “Yet another markup language”
- Metadata of your document
- Starts and ends with `---`
- Uses key-value pairs in the format `key: value`

Anatomy of a Quarto document

Text: Markdown

```
my-document.qmd
```

```
1 ---
2 title: "My Cool Document"
3 format: html
4 ---
5
6 This analysis explores data from the Gapminder foundation.
```

- Markdown is a lightweight language for creating formatted text
- Quarto is based on Pandoc and uses its variation of markdown as its underlying document syntax

Anatomy of a Quarto document

Text: Markdown

```
my-document.qmd
```

- 1 The `gapminder.csv` dataset contains data from
- 2 the [**Gapminder foundation**](<https://www.gapminder.org/>).



The `gapminder.csv` dataset contains data from the
Gapminder foundation.

Anatomy of a Quarto document

Text: Markdown

Markdown syntax

italics and **bold**

superscript² / subscript₂

~~strikethrough~~

`verbatim code`

Output

italics and **bold**

superscript² /
subscript₂

~~strikethrough~~

verbatim code

General Markdown guide

Anatomy of a Quarto document

Code

```
my-document.qmd
1 ---
2 title: "My Cool Document"
3 format: html
4 ---
5
6 The `gapminder.csv` dataset contains data from the [**Gapminder foundation*]
7
8 ````{r}
9 library(tidyverse)
10
11 df <- read_csv("data/gapminder.csv")
12
13 ggplot(df, aes(x = gdpPercap, y = lifeExp)) +
14   geom_point(aes(size = pop, color = continent)) +
15   scale_x_log10()
16 ````
```

Anatomy of a Quarto document

Code

```
my-document.qmd
1 ---
2 title: "My Cool Document"
3 format: html
4 ---
5
6 The `gapminder.csv` dataset contains
7
8 ```{r}
9 library(tidyverse)
10
11 df <- read_csv("data/gapminder.csv")
12
13 ggplot(df, aes(x = gdpPercap, y = lifeExp))
14   geom_point(aes(size = pop, color = continent))
15   scale_x_log10()
16 ```


```

- Code chunks begin and end with three backticks
- Code chunks are identified with a programming language in between {}

Anatomy of a Quarto document

Inline code executes code *within* Markdown

```
my-document.qmd
```

```
1 ``{r}
2 countries <- 147
3
4
5 There are `r countries` in the dataset.
```



There are 147 countries in the dataset.

Anatomy of a Quarto document

Code can include optional chunk options, in YAML style, identified by `#|` at the beginning of the line

```
my-document.qmd
1 The `gapminder.csv` dataset contains data from the [**Gapminder foundation*
2
3 ````{r}
4 #| label: fig-neat-plot
5 #| echo: false
6 #| fig-width: 6
7 #| fig-height: 3.8
8 #| fig-cap: "My neat plot"
9
10 library(tidyverse)
11
12 df <- read_csv("data/gapminder.csv")
13
14 ggplot(df, aes(x = gdpPercap, y = lifeExp)) +
15   geom_point(aes(size = pop, color = continent)) +
16   scale_x_log10()
17 ````
```

Anatomy of a Quarto document

Code can include optional chunk options, in YAML style, identified by `#|` at the beginning of the line

Option	Description
<code>eval</code>	Evaluate the code chunk
<code>echo</code>	Include the source code
<code>warning</code>	Include warnings
<code>include</code>	Include code and results

Other chunk options

Your turn

- Open `01-exercise.qmd` and run some of the code chunks (in order!).
- Add `#| include: false` to the second chunk and preview again. Switch it back to `true` or remove it. Preview again.
- In the YAML area, add an `author` field and add your name. Preview again.
- Change `code-fold` to be true. Preview again.
- Edit the first paragraph to:
 1. Make something bold
 2. Make “the Gapminder foundation” link to <https://www.gapminder.org>
 3. Replace `xxxx` and `YYYY` with inline code instead of hardcoded values. The first code chunk creates R objects named `first_year` and `last_year`—use those.

05:00

Authoring Quarto

Images and links

Markdown syntax

<<https://quarto.org>>

[Quarto](<https://quarto.org>)

Output

<https://quarto.org>

Quarto



Tables

1	Right	Left	Default	Center	
2					
3	12	12	12	12	
4	123	123	123	123	
5	1	1	1	1	



Right Left Default Center

12	12	12	12
123	123	123	123
1	1	1	1

Tables

```
1 | Right | Left | Default | Center |
2 |-----:|:-----|-----:|-----:|
3 |     12 | 12   | 12    | 12    |
4 | 123  | 123  | 123   | 123   |
5 |     1  | 1    | 1     | 1     |
6
7 : Table Column Widths {tbl-colwidths="[10,30,30,30]"}  
↓
```

Table Column Widths

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

Citations

```
my-document.qmd
```

```
1 ---  
2 title: "My Cool Document"  
3 format: html  
4 bibliography: references.bib  
5 ---  
6  
7 Computers are neat [@Lovelace:1842].
```



Computers are neat (Lovelace 1842).

References

Lovelace, Ada Augusta. 1842. "Sketch of the Analytical Engine Invented by Charles Babbage, by LF Menabrea, Officer of the Military Engineers, with Notes Upon the Memoir by the Translator." *Taylor's Scientific Memoirs* 3: 666–731.

Citations

```
my-document.qmd
```

```
1 ---  
2 title: "My Cool Document"  
3 format: html  
4 bibliography: references.bib  
5 csl: apa.csl  
6 ---  
7  
8 Computers are neat [@Lovelace:1842].
```



Computers are neat (Lovelace, 1842).

References

Lovelace, A. A. (1842). Sketch of the analytical engine invented by Charles Babbage, by LF Menabrea, officer of the military engineers, with notes upon the memoir by the translator. *Taylor's Scientific Memoirs*, 3, 666–731.

Citations

```
my-document.qmd
```

```
1 ---  
2 title: "My Cool Document"  
3 format: html  
4 bibliography: references.bib  
5 csl: chicago-notes-bibliography.csl  
6 ---  
7  
8 Computers are neat [@Lovelace:1842].
```



Computers are neat¹.

Footnotes

1. Ada Augusta Lovelace, "Sketch of the Analytical Engine Invented by Charles Babbage, by LF Menabrea, Officer of the Military Engineers, with Notes Upon the Memoir by the Translator," *Taylor's Scientific Memoirs* 3 (1842): 666-731. ↵

Citations

Zotero + Better BibTeX
can manage references
and export them for
Quarto

10,000 bibliographic styles
are available at
<https://www.zotero.org/styles>



Don't memorize this stuff!

The screenshot shows a navigation bar with links for Overview, Get Started, Guide, Extensions, Reference, Gallery, Blog, Help, social media icons, and a search bar. A 'supported by posit' logo is also present. The main content area has a breadcrumb trail: Guide > Authoring > Scholarly Writing > Citations. The left sidebar contains a hierarchical menu for 'Authoring' and 'Computations'. The right sidebar lists 'On this page' topics like Overview, Bibliography Files, Citation Syntax, etc. The main content starts with a section titled 'Citations' and 'Overview', followed by a note about Typst citation processing, and ends with a section on 'Bibliography Files'.

Guide > Authoring > Scholarly Writing > Citations

Citations

Overview

Quarto will use Pandoc to automatically generate citations and a bibliography in a number of styles. To use this capability, you will need:

- A quarto document formatted with citations (see [Citation Markdown](#)).
- A bibliographic data source, for example a BibLaTeX (`.bib`) or BibTeX (`.bibtex`) file.
- Optionally, a `CSL` file which specifies the formatting to use when generating the citations and bibliography (when not using `natbib` or `biblatex` to generate the bibliography).

Note

When using `format: typst`, by default citation processing is handled by Typst, not Pandoc. See the [Typst](#) section below for more details.

Bibliography Files

Quarto supports bibliography files in a wide variety of formats including BibLaTeX and CSL. Add a bibliography to your document using the [bibliography YAML](#).

Quarto.org > Guide > Authoring > Scholarly Writing > Citations

Cross references

```
my-document.qmd
```

```
1 See @fig-neat-plot for more details.  
2  
3 ````{r}  
4 #| label: fig-neat-plot  
5 #| fig-cap: "My neat plot"  
6  
7 # Plot code here  
8 ````
```



See Figure 1 for more details.

...

Figure 1: My neat plot

Don't memorize this stuff!



Overview Get Started Guide Extensions Reference Gallery Blog Help ▾



supported by

Guide
Authoring
Markdown Basics
Figures
Tables
Diagrams
Shortcodes
Videos
Embeds
Callout Blocks
Code Annotation
Brand
Article Layout
Scholarly Writing
Front Matter
Title Blocks
Citations
Cross-References
Basics
Options
Div Syntax
Custom Floats
Creating Citeable Articles
Appendices
Computations

Guide > Authoring > Scholarly Writing > Cross-References > Basics

Cross References

Overview

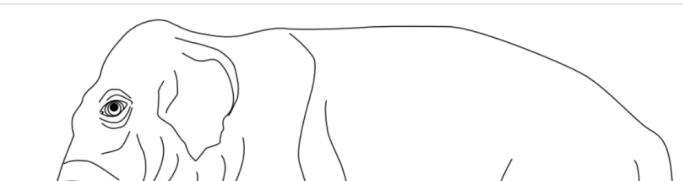
Cross-references make it easier for readers to navigate your document by providing numbered references and hyperlinks to various entities like figures and tables. Every cross-referenceable entity requires a label—a unique identifier prefixed with a cross-reference type e.g. `#fig-element`. For example, this is a cross-referenceable figure:

```
! [Elephant] (elephant.png) {#fig-elephant}
```

The presence of the label `(#fig-elephant)` makes this figure referenceable. This enables you to use the following syntax to refer to it elsewhere in the document:

```
See @fig-elephant for an illustration.
```

Here is what this would look like rendered to HTML:



On this page
Overview
References
Lists
Floats
Figures
Tables
Code Listings
Callouts
Theorems and Proofs
Equations
Sections

>Edit this page
Report an issue

Quarto.org > Guide > Authoring > Scholarly Writing > Cross-References

Divs and Spans

For further customization, you can add classes, attributes, and other identifiers to content using divs and spans.

Divs

```
1 ::: {.border}
2 This adds the "border" class to some content.
3 :::
```

Spans

```
1 [This is some text]{.class style="color: #cccccc;"}
```

Divs

Callout blocks

```
my-document.qmd
```

```
1 :::{.callout-tip}
2
3 Note that there are five types of callouts, including:
4 `note`, `tip`, `warning`, `caution`, and `important`.
5
6 :::
```



Tip

Note that there are five types of callouts, including: `note`, `tip`, `warning`, `caution`, and `important`.

Divs

Multiple columns

```
my-document.qmd
```

```
1 ::: {layout-ncol=2}
2
3 
4
5 Photo by [The New York Public Library](https://unsplash.com/@nypl) on [Unsp
6 :::
```

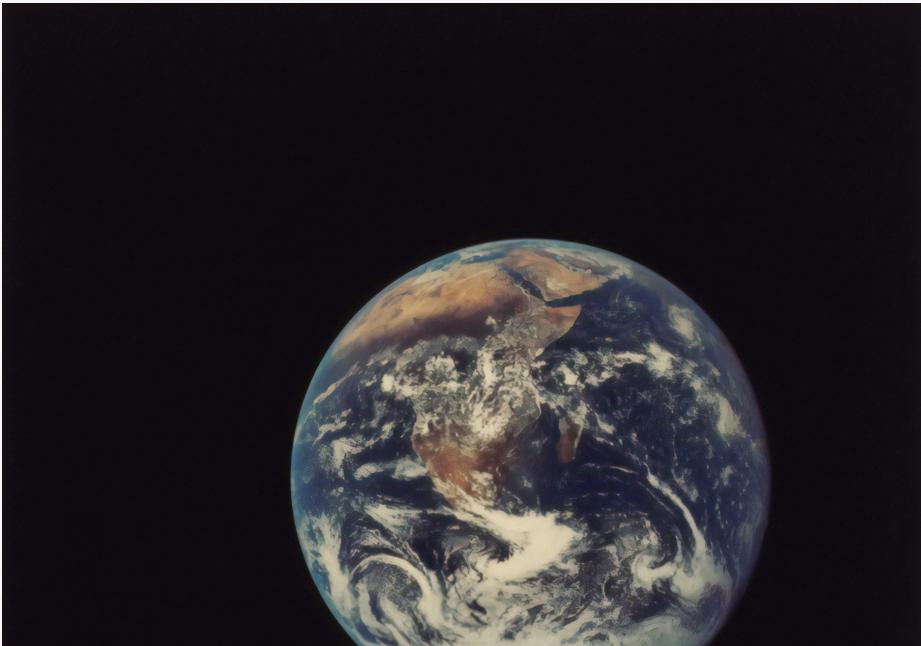


Photo by **The New York
Public Library** on
Unsplash

Divs

Tabs

my-document.qmd

```
1 ::: {.panel-tabset}
2 ## R
3
4 `library(dplyr)`
5
6 ## Python
7
8 `import pandas as pd`
9 :::
```

R

Python

library(dplyr)

Spans

```
my-document.qmd
```

```
1 This is text that is [red]{style="color:red;"}.
```

This is text that is red.

Changing formats

```
my-document.qmd
```

```
1  ---
2  title: "My Cool Document"
3  format: html
4  ---
```

Changing formats

```
my-document.qmd
```

```
1 ---  
2 title: "My Cool Document"  
3 format: revealjs  
4 ---
```

Health and Wealth Around the World

Gapminder data, 1952–2007

2025-10-16



Don't memorize this stuff!

 quarto Overview Get Started Guide Extensions Reference Gallery Blog Help ▾

[Twitter](#) [GitHub](#) [RSS](#) [Open in new window](#) [Search](#) 

Guide > Authoring > Markdown Basics

Markdown Basics

Overview

Quarto is based on Pandoc and uses its variation of markdown as its underlying document syntax. Pandoc markdown is an extended and slightly revised version of John Gruber's [Markdown](#) syntax.

Markdown is a plain text format that is designed to be easy to write, and, even more importantly, easy to read:

A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – [John Gruber](#)

This document provides examples of the most commonly used markdown syntax. See the full documentation of [Pandoc's Markdown](#) for more in-depth documentation.

Text Formatting

Markdown Syntax	Output
<code>*italics*, **bold**, ***bold italics***</code>	<i>italics</i> , bold , <i>bold italics</i>
<code>superscript^2^ / subscript~2~</code>	$\text{superscript}^2 / \text{subscript}_2$

On this page

- Overview
- Text Formatting
- Headings
- Links & Images
- Lists
- Footnotes
- Tables
- Source Code
- Raw Content
- Equations
- Diagrams
- Videos
- Page Breaks
- Divs and Spans
- Callout Blocks
- Other Blocks
- Other Spans
- Special Characters
- Keyboard Shortcuts

[Edit this page](#) [Report an issue](#)

Your turn

- Add an [important](#) callout box to the introduction summarizing the report's findings. Preview the file.
- Change the caption for the first plot. Preview again.
- Change some text color to [#ec008b](#). Preview again.
- Edit the "Average life expectancy increased..." paragraph to say "According to Figure 1, average life expectancy increased...", **BUT** do it without typing "Figure 1". Preview again.
- In the first paragraph, add a citation to something in [bib/references.bib](#). Preview again.
- Change the bibliography style to APA. Preview again.

05 : 00

What's next?

Course outline

-  ~~Intro to Quarto~~
- Creating basic websites
- Advanced website features
- Publishing
- Customization and branding
- Interactivity