

Creating Reproducible Documents with Quarto



Lubov McKone

lmckone1@jh.edu

Johns Hopkins Sheridan Libraries - Data Services

About Data Services

We help faculty, researchers, and students find, use, manage, visualize and share data.

<https://dataservices.library.jhu.edu/>

dataservices@jhu.edu

About you

Using the Zoom  reaction,

- Have you used Quarto before?
- Have you used RMarkdown before?

About you

Please open the [etherpad](#) (link in the chat), and answer the following questions

1. Do you use a programming language (such as R, Python, etc.) for your research? If so, which one?
2. What is bringing you to this training? What kinds of documents or outputs are you hoping to use Quarto to create?

Today

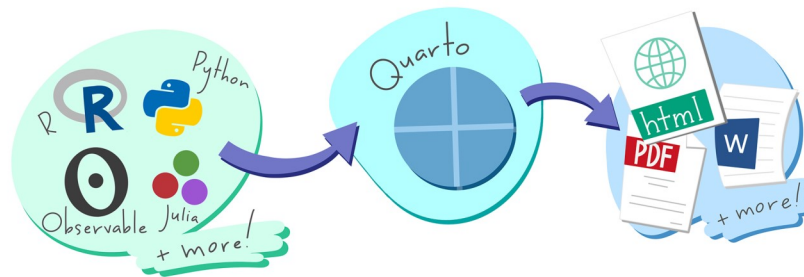
- Background
- Quarto tour
- **Hands-on portion:** Creating documents & presentations
- **Demo portion:** Creating manuscripts & beyond
- Resources & further learning



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

Quarto overview

- Quarto is an **authoring framework** for data science.
- It is a tool that allows you to create a **variety of output documents** (HTML, PDF, MS Word, MS PowerPoint, and more!) from a combination of **text and code** (in R, Python, Julia, and more).



Like this

Here's an **example**

Introduction

This is a Quarto document. To learn more about Quarto see <https://quarto.org>.

Click the **Code** button in the header to see the full source code of this document.

Here we call the R `summary()` function—the function's output is included immediately below:

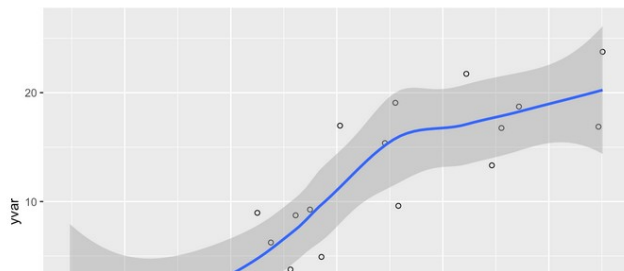
```
summary(cars)
```

	speed	dist
Min. :	4.0	Min. : 2.00
1st Qu.:12.0		1st Qu.: 26.00
Median :15.0		Median : 36.00
Mean :15.4		Mean : 42.98
3rd Qu.:19.0		3rd Qu.: 56.00
Max. :25.0		Max. :120.00

Plot Output

You can also embed plots, for example:

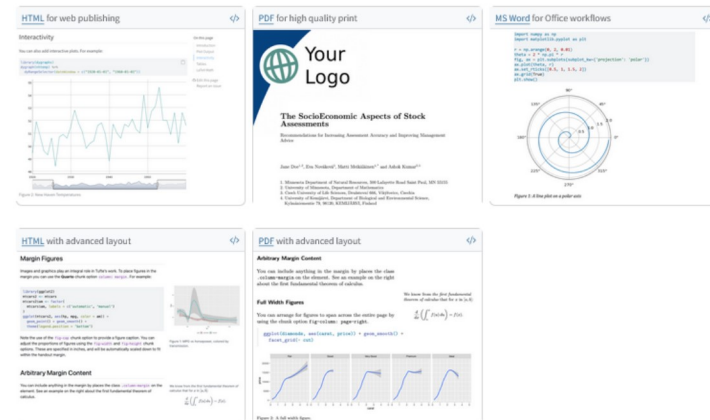
► Code



Here's **even more examples**

Articles & Reports

Write a single markdown file and create output in a variety of formats.



Presentations

Create presentations (slide show) in a variety of formats.



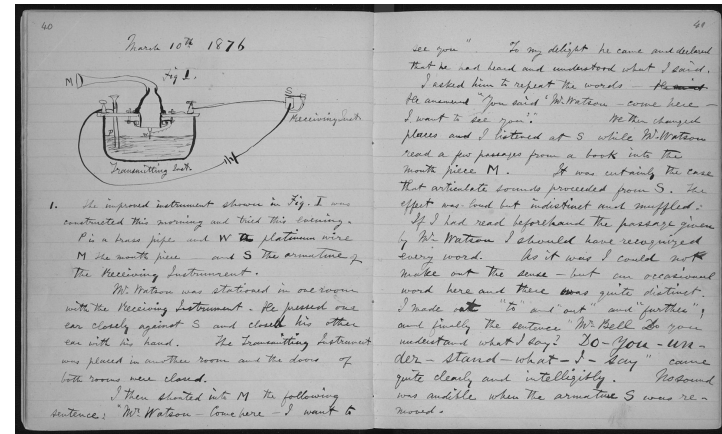
Quarto as next-generation R Markdown

- Quarto unifies and extends the R Markdown ecosystem
 - unifies for people who love R Markdown and all its spinoffs
 - extends for people who don't know R Markdown



Quarto: Why?

- We can think of *notebooks* as documents that mix prose, code, and output.
- Providing notebooks as curated research outputs would greatly enhance transparency and reproducibility.



Page from a laboratory notebook of Alexander Graham Bell, 1876.

Quarto: How?

- Quarto documents can be edited and rendered in **RStudio**, **Jupyter Notebooks**, and **Visual Studio**, and via the **Quarto Command Line Interface**.
- What happens when a Quarto document is rendered:



Quarto engines

- Quarto engines do two things:
 1. Execute code
 2. Render text, code, and output into markdown (“make it pretty”)
- Quarto’s current engines (knitr and Jupyter) can execute R, Python, Julia, ObservableJS, and many more languages.
- You can select an engine or R will automatically detect which engine to use based on your document. Functionally, the engines are extremely similar, and are *almost* identical in syntax

Quarto engines

knitr

- Renders R and Julia natively.
- Renders python via the [reticulate](#) package
- Comes installed with RStudio.
- Best for Quarto beginners or RMarkdown converts.

Jupyter

- Renders R, Python, Julia, and many more languages natively.
- Requires installation of Jupyter, in addition to any other kernels you might.
- Best for those who are familiar with Jupyter.

Quarto tour

- When you create a Quarto document, you will see options for many different type of output documents, and a place to designate the engine you would like to use.
 - We recommend not setting an engine. This will allow Quarto to dynamically select an engine based on the content you have in your document.
- The YAML header controls the document format and is automatically generated based on options you select when creating your document.
 - There are thousands of YAML header options that vary by output type, a good reference [here](#).















Text in documents

- Create headings and text by typing in the visual editor
- Insert citations by DOI, crossref search, or from your Zotero library with [Insert > Citation](#) or with [@](#)
- Insert footnotes with [Insert > Footnote](#)
- Add tabsets to separate components of your analysis with [Insert > Tabset](#)

Code in documents

- Insert executable code with `Insert > Executable Cell`
 - you can combine programming languages in one document
- Specify options for displaying code chunks with `#|` at the beginning of the code chunk (see next slide).
- Specify options for displaying code chunks globally in the YAML header with `execute:` followed by the option.

Code execution options

when set to false (i.e. <code>eval:</code> <code>false</code>)	run code	show code	show output	show warning	show errors
<code>eval</code>					
<code>echo</code>					
<code>output</code>					
<code>include</code>					
<code>warning</code>					
<code>error</code>					

Other code options

- You can add annotations to your code by inserting a number tag, like `# <1>`, next to the line of code you want to annotate. Then, immediate under the code chunk, add the number followed by the annotation, like `1. [Insert annotation text]`
- To allow users to expand or hide code chunks, you can add the option `code-fold: true` under `format: html:` in the YAML header, or as an options for a specific code chunk with `#| code-fold: true`

Inline code

- You can add inline code to your document text with the syntax `{r} 2 + 2` surrounded by backticks `
- The above will render in your Markdown to 4

Using Python and R together

- Quarto can execute Python and R together in the same document with the `reticulate` R package
- When you add a Python code chunk to a Quarto document, R will ask if you want to create a `reticulate` environment for the document. This will manage the Python version and packages you have access to in the document.
- Once you do this, you will see that you now have a Python Console and Environment in addition to R.
- You can reference R objects in Python with `r.`, and vice versa with `py$`.

Referencing outputs in documents

- You can make figures produced by code chunks can be referenced from elsewhere in your document by specifying the code chunk option `#| label:` and giving it a label with the prefix `fig-` .
- The output can then be referenced in your document text with `@fig-label` .
- Images, tables, and numbers can also be referenced, but the syntax differs - see [here](#) for a full reference.

Slides in Quarto

- One of the nicest things about Quarto is that you can easily convert between output formats
 - Go from document to slides by changing `format: html` to `format: revealjs` in the YAML header
 - Slides are delineated by headings
- Create Quarto slides from scratch with `New File > Quarto Presentation`

Quarto slides features

- Add slide notes with `Insert > Slide Notes`
- Speaker View allows for easy presenting, and PDF export mode allows for easy sharing
- You can add the ability to annotate slides with `format: revealjs: chalkboard: true`

Quarto Dashboards

- Like Shiny, but with a focus on layout and organization of components without native reactivity or interactivity.
- Interactivity can be added to dashboards with Shiny components.
- See this presentation on [Quarto Dashboards](#) for more!

Quarto manuscripts

- Let's explore a Quarto [manuscript](#)
- In RStudio:
 - Create a manuscript with New Project > New Directory > Quarto Manuscript
 - Add new journal format in the Terminal window with `quarto add quarto-journals/plos`, full list of journal formats [here](#), and change outputs in `_quarto.yml`

Referencing external notebooks

- Within your manuscript, create a folder called **notebooks**
- Insert any .qmd or .ipynb notebooks that you'd like to reference in your manuscript
- You can embed code and/or output in your manuscript without running it every time you render your manuscript, go to Insert > Shortcode and type `embed notebooks/ NotebookName.qmd#cell-label`



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

Further learning

- [Reproducible Manuscripts with Quarto, Mine Çetinkaya-Rundel, posit::conf\(2023\)](#)
- [Reproducible Manuscripts with Quarto, JJ Allaire, Bioconductor](#)
- [Custom CSS for Quarto with Sass](#)
- [Customizing figures in Quarto](#)
- [Using parameters in Quarto documents](#)

thank you! 🙏

Please let us know what you thought of this workshop at bit.ly/quarto-survey

Any questions?

Upcoming workshops:

- Data Bytes: The Hidden Mapping Powers of ArcGIS Arcade: 11/18 12-1pm
- Finding a Repository to Share Research Data: 12/3 12-1pm