



A brief introduction to literate programming using Quarto

Mutaz Jaber, PharmD, PhD mutaz.jaber@gilead.com

Emma Hughes, PhD emma.hughes2@gilead.com

Materials: <https://github.com/Mutaz94/quarto-CTSI-2024>

13th Annual Indiana CTSI Therapeutic Response Modeling Program Symposium
Indianapolis, IN
Feb 28, 2024

Credit is due

Many thanks to

- Devin Pastoor, PhD (A2-AI)
- Ana Ruiz, PharmD, PhD (Gilead Sciences, Inc)
- Nieves Velez de Mendizabal, PhD (Gilead Sciences, Inc)
- CTSI Staff and members

Disclosure



We are not affiliated with Posit Co or the development of Quarto software. Our purpose is to provide information and instruction to use the software. The views and information presented here are those of the authors and not Gilead Sciences Inc.

Literate Programming

Donald E. Knuth

Computer Science Department, Stanford University, Stanford, CA 94305, USA

The author and his associates have been experimenting for the past several years with a programming language and documentation system called WEB. This paper presents WEB by example, and discusses why the new system appears to be an improvement over previous ones.

“Literate programming is a methodology that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language.”

Beyond documentation, literate programming offers benefits in education, research, and various technical domains.

- Manuscripts
- Technical Reports
- Academic theses
- Documentation
- Class material
- Many others



A Quarto Page Layout Example

Inspired by Tufte Handout, Using Quarto

2023-04-30

Introduction

This document demonstrates the use of a number of advanced page layout features to produce an attractive and usable document inspired by the Tufte handout style and the use of Tufte's styles in RMarkdown documents (Xie, Allaire, and Grolemund 2018). The Tufte handout style is a style that Edward Tufte uses in his books and handouts. Tufte's style is known for its extensive use of sidenotes, tight integration of graphics with text, and well-set typography. Quarto¹ supports most of the layout techniques that are used in the Tufte handout style for both HTML and LaTeX/PDF output.

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. "Tufte Handouts." In *R Markdown: The Definitive Guide*, 137–46. Chapman; Hall/CRC. <https://doi.org/10.1201/978133394446>.

¹ To learn more, you can read more about Quarto or visit Quarto's Github repository.

```
---
```

```
title: "An Example Using the Tufte Style"
author: "John Smith"
format:
  html:
    grid:
      margin-width: 350px
    pdf: default
    reference-location: margin
    citation-location: margin
  ---
```

^① Increases the width of the margin to make more room for sidenotes and margin figures (HTML only).

We use it in our pharmacometric analysis workflow to create reports and presentations

POPULATION PHARMACOKINETICS REPORT

Report Number:	QP12345
Report Title:	A Big Title for Study A and B
Study Drug:	GS-12345
Indication(s):	
Study Number(s):	GS-US-123-4567 GS-US-123-4568
Sponsor:	Gilead Sciences, Inc. 333 Lakeside Drive Foster City, CA 94404 USA
Prepared By:	Author 1 Author 2 Author 3
Reviewed By:	Reviewer 1 Reviewer 2 Reviewer 3
Approved By:	Approve 1
Report Date:	02 Feb 2024
Report Status:	Draft

CONFIDENTIAL

This is a Gilead Sciences, Inc. document that contains confidential information. It is intended solely for the recipient and must not be disclosed to any other party. This material may be used only for evaluating or conducting clinical investigations; any other proposed use requires written consent from Gilead Sciences, Inc.

GS-12345 Weekly Meetings

My Awesome Pharmacometrics Project Title

Author Author

June 15, 2023

All Rights Reserved ©2023

 GILEAD | Clinical Pharmacology Sciences

Introduction to Quarto

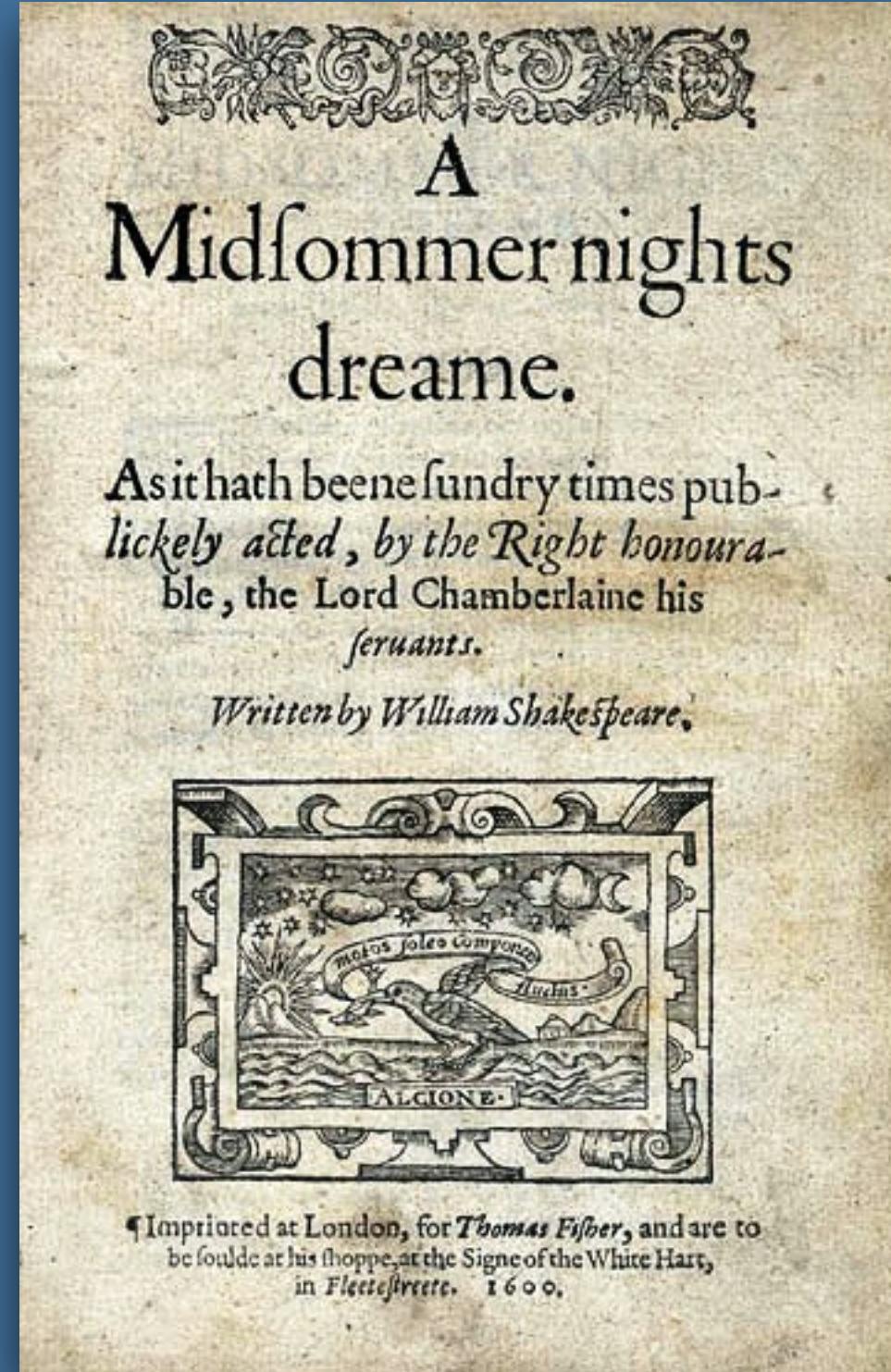
ACoP14 Workshop Session

Emma Huges and Ana Ruiz

2023-11-04

 GILEAD Creating Possible 

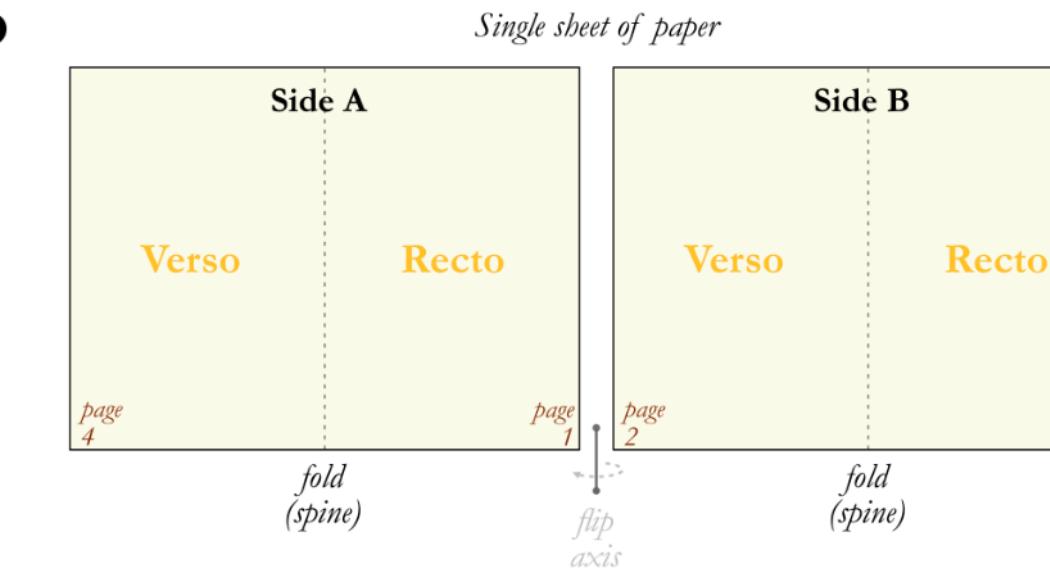

Hello, Quarto!



Wikipedia: Title page of the **first quarto edition** of Shakespeare's *Midsummer Night's Dream*, 1600, from the Folger Shakespeare Library

FOLIO

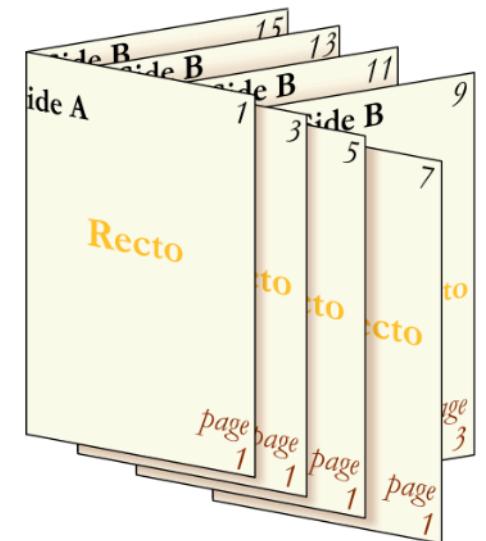
abbreviated
fo 2°



- 4 pages of printed text
- 2 on each side
- 1 fold
- creates 2 leaves
- each leaf is 1/2 size of the original sheet

<https://en.wikipedia.org/wiki/Folio>

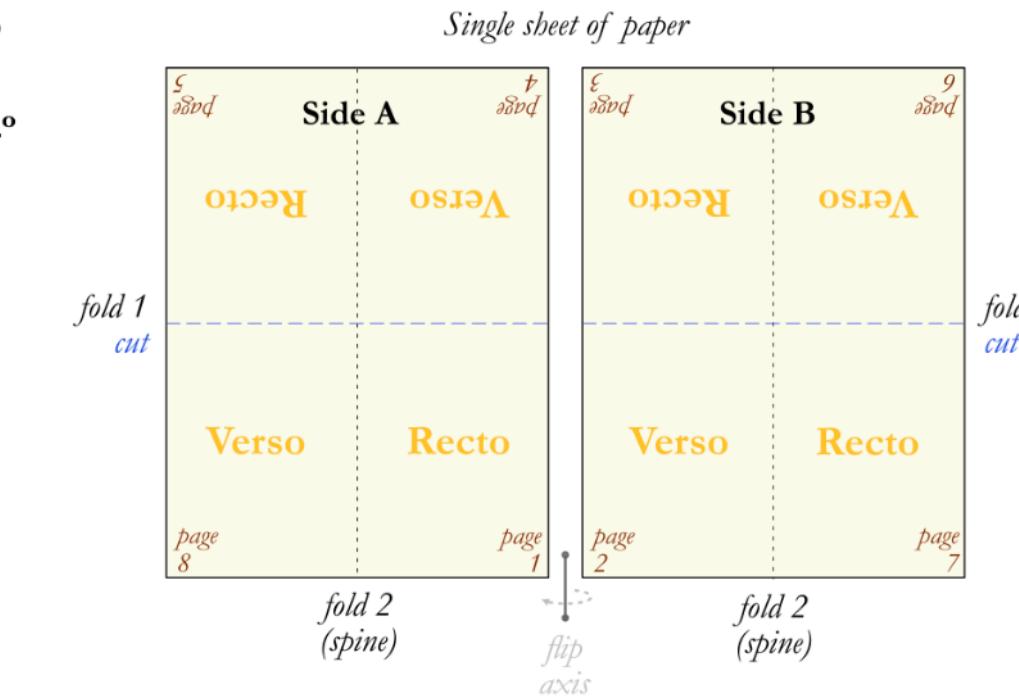
Example of a Folio Gathering



Folios would be inserted within each other to form, for example, gatherings of 8 leaves (above). Gatherings are sewn through the central fold prior to binding the book. Multiple gatherings are stacked and made into a book.

QUARTO

abbreviated
Qto, 4to or 4°

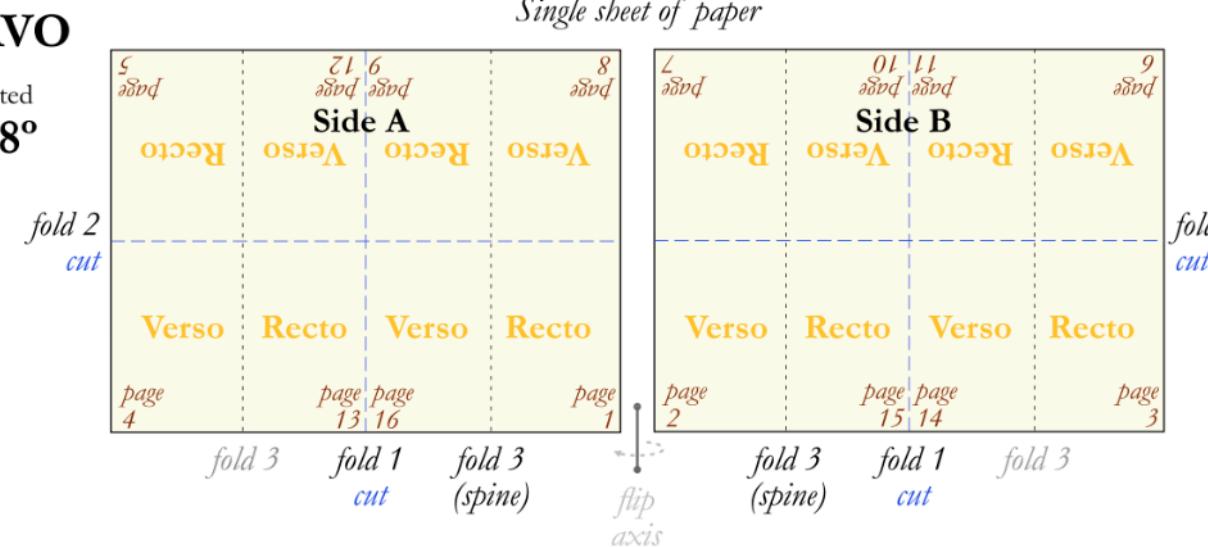


- 8 pages of printed text
- 4 on each side
- 2 folds, 1 cut
- creates 4 leaves
- each leaf is 1/4 size of the original sheet

<https://en.wikipedia.org/wiki/Quarto>

OCTAVO

abbreviated
8vo, 8°



- 16 pages of printed text
- 8 on each side
- 3 folds, 2 cuts
- creates 8 leaves
- each leaf is 1/8 size of the original sheet

<https://en.wikipedia.org/wiki/Octavo>

For "left-to-right reading" books.

Hello, Modern Quarto: www.quarto.org/

The screenshot shows the Quarto website homepage. At the top, there's a navigation bar with the Quarto logo, Overview, Get Started, Guide, Extensions, Reference, Gallery, Blog, and Help. Below the navigation is a banner announcing "Quarto 1.4 released! Download, Read More" with a small party popper icon. The main content area features a large blue header "Welcome to Quarto" followed by a subheader "An open-source scientific and technical publishing system". A bulleted list details the system's capabilities: authoring with Jupyter notebooks, creating dynamic content with Python, R, Julia, and Observable, publishing reproducible content in various formats, sharing knowledge organization-wide, and writing using Pandoc markdown. To the right, a sidebar titled "Publishing Options" lists "Websites / Blogs" (with a Jupyter icon), "Dashboards" (with a chart icon), "Articles" (with a document icon), "Presentations" (with a slide icon), "Books" (with a book icon), and "Knowledge Repos" (with a lightbulb icon). At the bottom, there are two buttons: "Get Started" (blue) and "Guide" (grey).

Quarto 1.4 released! [Download](#), [Read More](#) X

Welcome to Quarto®

An open-source scientific and technical publishing system

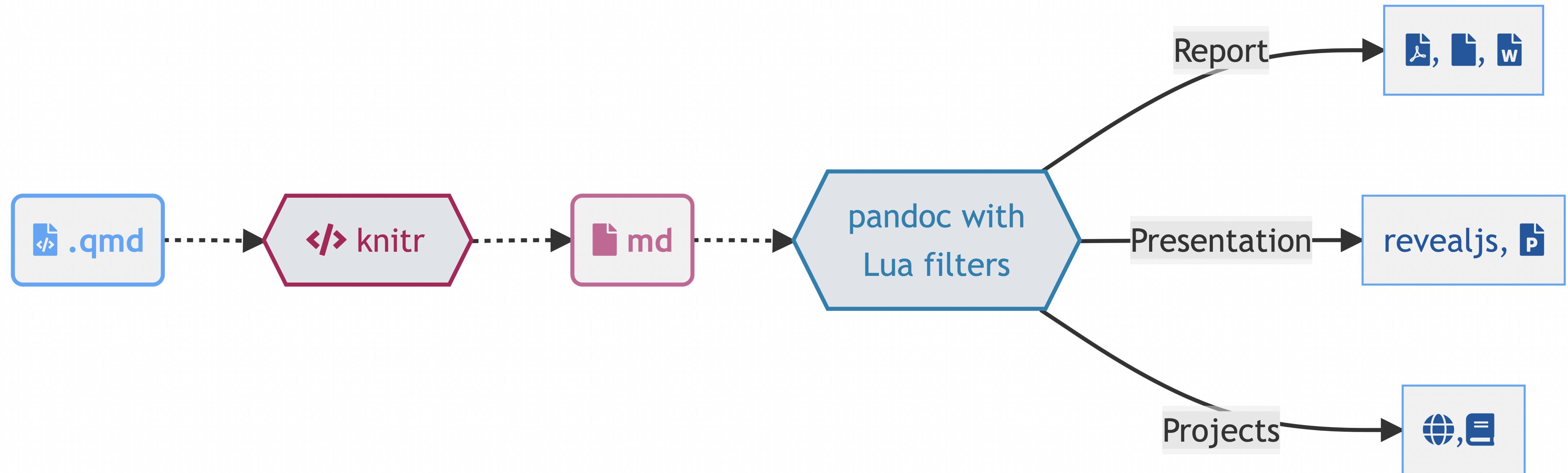
- Author using [Jupyter](#) notebooks or with plain text markdown in your favorite editor.
- Create dynamic content with [Python](#), [R](#), [Julia](#), and [Observable](#).
- Publish reproducible, production quality articles, presentations, dashboards, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
- Share knowledge and insights organization-wide by publishing to [Posit Connect](#), [Confluence](#), or other publishing systems.
- Write using [Pandoc](#) markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.

Analyze. Share. Reproduce. You have a story to tell with data—tell it with Quarto.

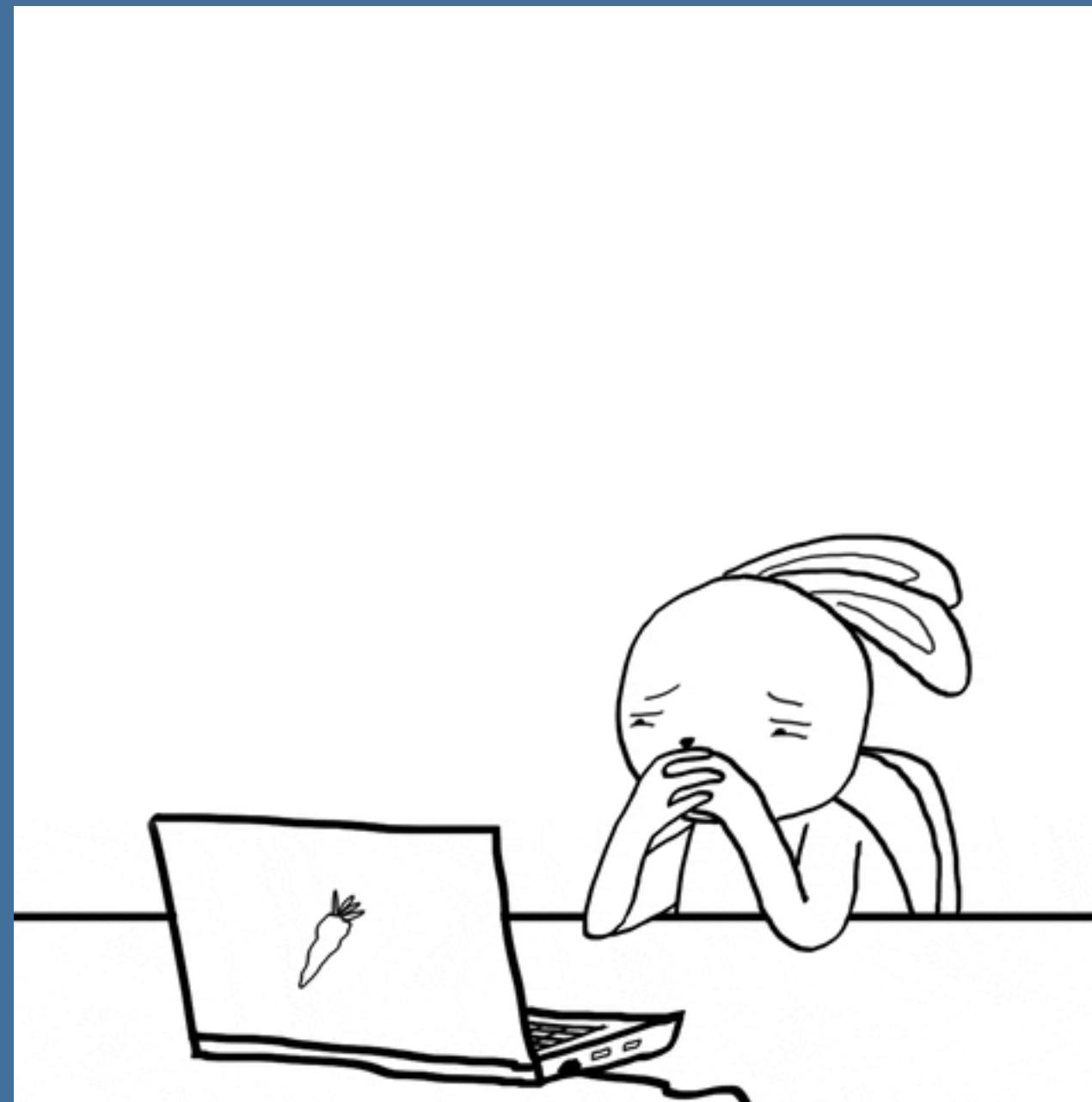
[Get Started](#) [Guide](#)

- Websites / Blogs
- Dashboards
- Articles
- Presentations
- Books
- Knowledge Repos

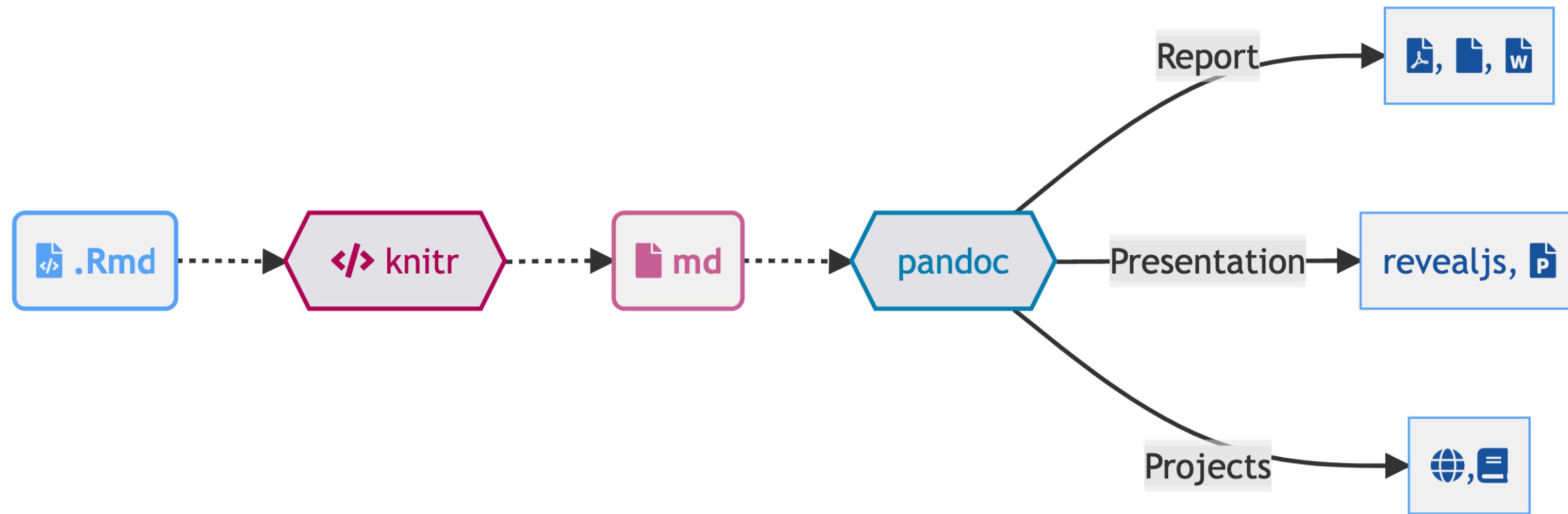
Quarto workflow



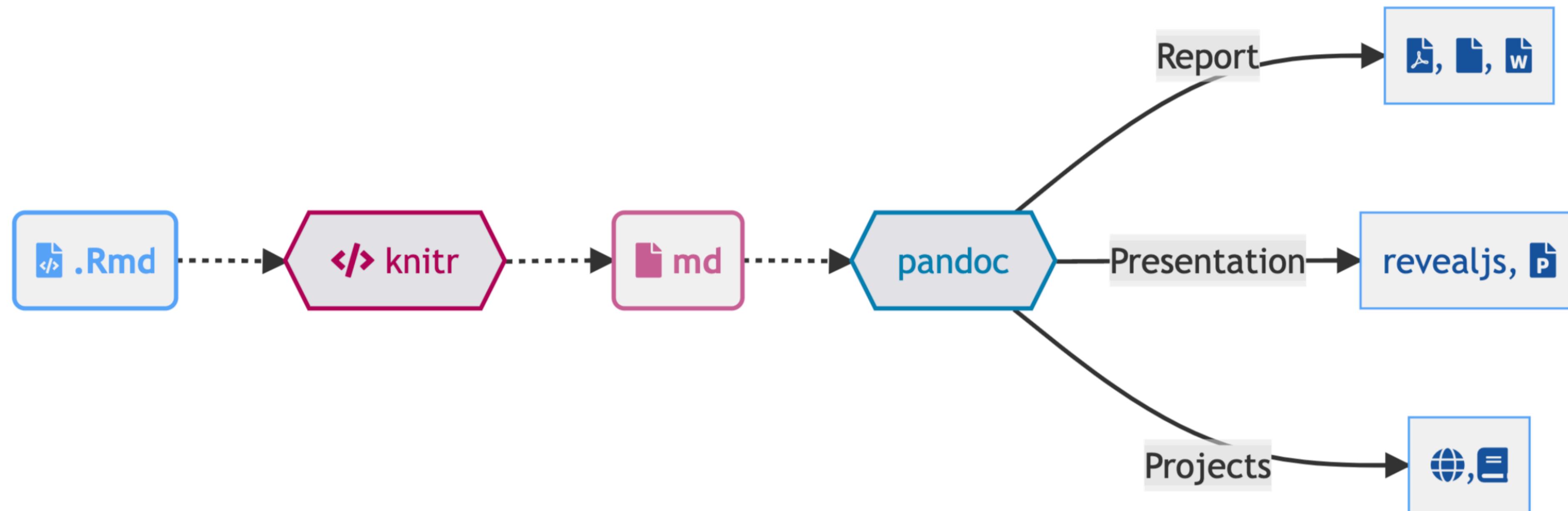
Wait! What about RMarkdown?



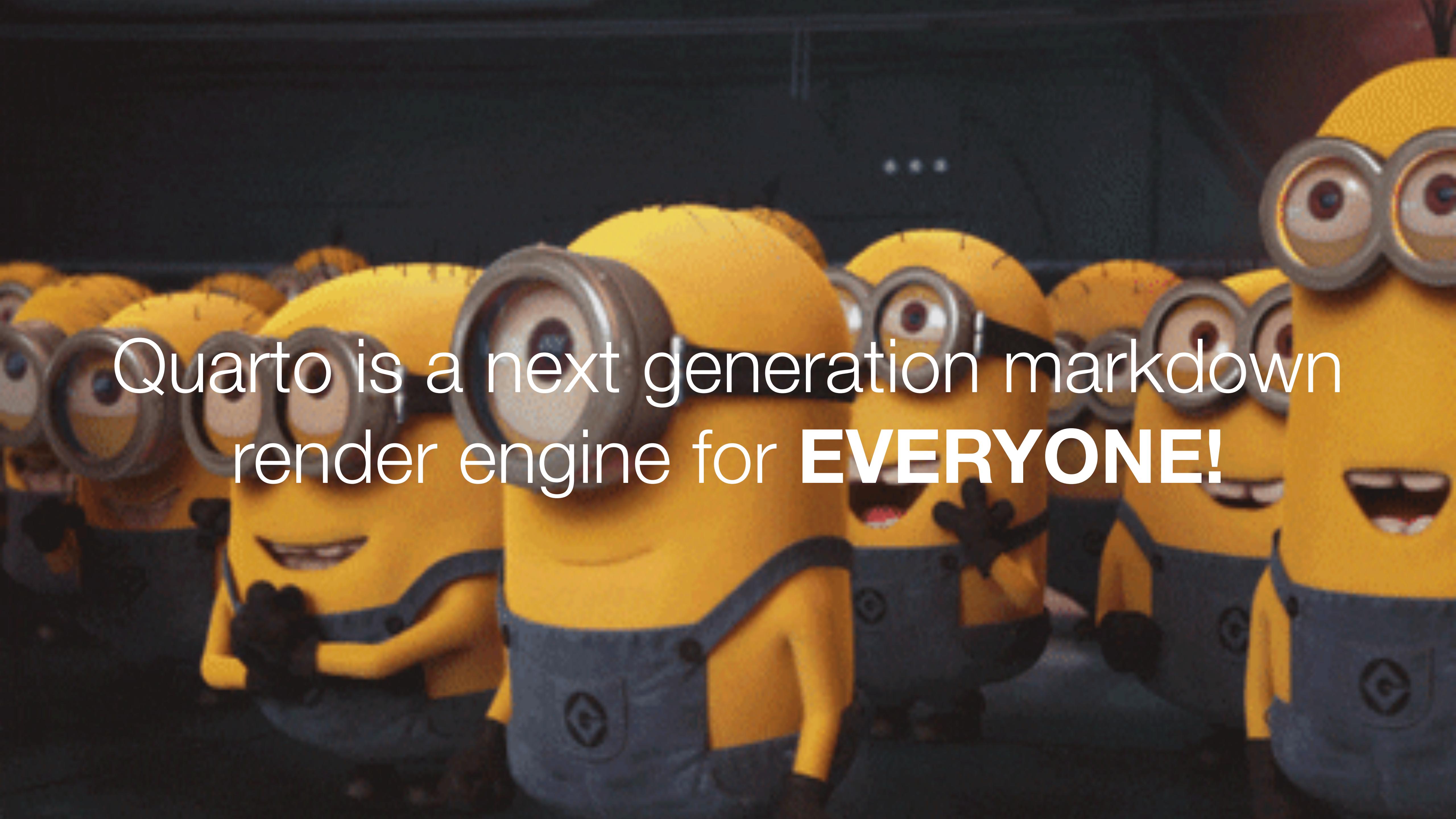
RMarkdown workflow



RMarkdown workflow



The goal of **Quarto** is to make the process of creating and collaborating on scientific and technical documents **dramatically better**. Quarto combines the functionality of R Markdown, bookdown, distill, xaringian, etc into a single consistent system with “batteries included” that reflects everything we’ve learned from R Markdown over the past 10 years.

A row of yellow Minions with black overalls and goggles, looking towards the right.

Quarto is a next generation markdown
render engine for **EVERYONE!**

Quarto is a command line (CLI) interface!

Quarto is a command line interface (CLI)
that renders plain text formats into
human readable/shareable formats

```
(base) ~ quarto --help

Usage: quarto
Version: 1.5.9

Description:
Quarto CLI

Options:
-h, --help      - Show this help.
-V, --version   - Show the version number for this program.

Commands:

render    [input] [args...]      - Render files or projects to various document types.
preview   [file] [args...]      - Render and preview a document or website project.
serve     [input]                - Serve a Shiny interactive document.
create    [type] [commands...]  - Create a Quarto project or extension
use       <type> [target]       - Automate document or project setup tasks.
add       <extension>          - Add an extension to this folder or project
update   [target...]           - Updates an extension or global dependency.
remove   [target...]           - Removes an extension.
convert   <input>              - Convert documents to alternate representations.
pandoc    [args...]            - Run the version of Pandoc embedded within Quarto.
typst     [args...]            - Run the version of Typst embedded within Quarto.
run      [script] [args...]    - Run a TypeScript, R, Python, or Lua script.
install   [target...]          - Installs a global dependency (TinyTex or Chromium).
uninstall [tool]               - Removes an extension.
tools     [provider] [path]     - Display the status of Quarto installed dependencies
publish   [provider] [path]     - Publish a document or project to a provider.
check    [target] [command]    - Verify correct functioning of Quarto installation.
help      [command]            - Show this help or the help of a sub-command.
```

Installing Quarto

<https://quarto.org/docs/get-started/>

The screenshot shows the Quarto website's 'Get Started' section. At the top, there's a banner for 'Quarto 1.4 released!' with links to 'Download' and 'Read More'. Below the banner, on the left, is a sidebar with 'Get Started' and three tutorial links: 'Tutorial: Hello, Quarto', 'Tutorial: Computations', and 'Tutorial: Authoring'. The main content area has a large heading 'Get Started' and a sub-instruction 'Install Quarto, then check out the tutorials to learn the basics.' A 'Step 1' section titled 'Install Quarto' includes a 'Download Quarto CLI' button for Mac OS. To the right is a table of download links for various platforms.

Platform	Download	Size	SHA-256
Ubuntu 18+/Debian 10+	quarto-1.4.550-linux-amd64.deb	111.83 MB	5ba82c8
Linux Arm64	quarto-1.4.550-linux-arm64.deb	112.53 MB	5d6a8cd
Mac OS	quarto-1.4.550-macos.pkg	186.21 MB	3c48831
Windows	quarto-1.4.550-win.msi	109.03 MB	18c6121

[Release notes and more downloads...](#)

Basic Usage: document anatomy

```
---
```

```
title: "Diamond sizes"
date: 2022-09-12
format: html
---
```

```
```{r}
#| label: setup
#| include: false
```

```
library(tidyverse)
```

```
smaller <- diamonds |>
 filter(carat <= 2.5)
```
```

We have data about `r nrow(diamonds)` diamonds.
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.
The distribution of the remainder is shown below:

```
```{r}
#| label: plot-smaller-diamonds
#| echo: false
```

```
smaller |>
 ggplot(aes(x = carat)) +
 geom_freqpoly(binwidth = 0.01)
```
```

Markdown flavored syntax!

Metadata Anatomy, a.k.a YAML

```
---
```

```
title: "Diamond sizes"
date: 2022-09-12
format: html
---
```

```
```{r}
#| label: setup
#| include: false

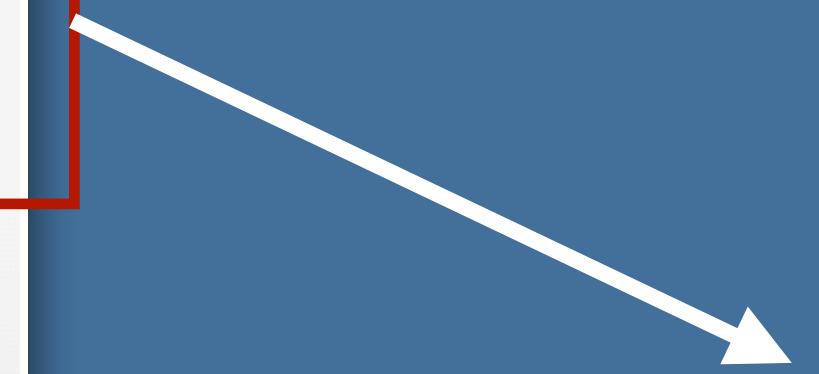
library(tidyverse)

smaller <- diamonds |>
 filter(carat <= 2.5)
```

We have data about `r nrow(diamonds)` diamonds.
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.
The distribution of the remainder is shown below:

```{r}
#| label: plot-smaller-diamonds
#| echo: false

smaller |>
 ggplot(aes(x = carat)) +
 geom_freqpoly(binwidth = 0.01)
```
```



Controls document settings
e.g., document info, formatting, and many others

Code chunk anatomy

```
---
```

```
title: "Diamond sizes"
date: 2022-09-12
format: html
---
```

```
```{r}
#| label: setup
#| include: false

library(tidyverse)

smaller <- diamonds |>
 filter(carat <= 2.5)
```
```

We have data about `r nrow(diamonds)` diamonds.
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.
The distribution of the remainder is shown below:

```
```{r}
#| label: plot-smaller-diamonds
#| echo: false

smaller |>
 ggplot(aes(x = carat)) +
 geom_freqpoly(binwidth = 0.01)
```
```



Contains:

- Chunk label: references
- Chunk options: appearance
- Source code: commands

Text (communication) anatomy

```
---
```

```
title: "Diamond sizes"
date: 2022-09-12
format: html
---
```

```
```{r}
#| label: setup
#| include: false
```

```
library(tidyverse)
```

```
smaller <- diamonds |>
 filter(carat <= 2.5)
```
```

We have data about `r nrow(diamonds)` diamonds.
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.
The distribution of the remainder is shown below:

```
```{r}
#| label: plot-smaller-diamonds
#| echo: false
```

```
smaller |>
 ggplot(aes(x = carat)) +
 geom_freqpoly(binwidth = 0.01)
```
```

Both; human readable communications and Source code

Rendering your first document

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the source code of `myFirstDoc.qmd`, which includes a title, a note about Quarto, a section on running code, and examples of executable code. The right pane shows the environment, which is currently empty. The bottom pane shows the terminal output of the command `quarto preview myFirstDoc.qmd --to html --no-watch-inputs --no-browse`, indicating the process is running.

```
myFirstDoc.qmd
1 ---  
2 title: "My First Document"  
3 format: html  
4 ---  
5  
6 ## Quarto  
7  
8 Quarto enables you to weave together content and executable code into a  
finished document. To learn more about Quarto see <https://quarto.org>.  
9  
10 ## Running Code  
11  
12 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:  
13  
14 ``{r}  
15 1 + 1  
16 ``  
17  
18 You can add options to executable code like this  
19  
20 ``{r}  
2:1 # My First Document
```

Environment is empty

Files Plots Packages Help Viewer Presentation

Preview: myFirstDoc.qmd Running 0:25

```
==> quarto preview myFirstDoc.qmd --to html --no-watch-inputs --no-browse

processing file: myFirstDoc.qmd
output file: myFirstDoc.knit.md

pandoc
  to: html
  output-file: myFirstDoc.html
  standalone: true
  section-divs: true
  html-math-method: mathjax
  wrap: none
  default-image-extension: png
```

Rendering your first document

The screenshot shows the RStudio interface with a Quarto document open. The document content is as follows:

```
1 ---  
2 title: "My First Document"  
3 format: html  
4 ---  
5  
6 ## Quarto  
7  
8 Quarto enables you to weave together content and executable code into a  
finished document. To learn more about Quarto see <https://quarto.org>.  
9  
10 ## Running Code  
11  
12 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:  
13  
14 ```{r}  
15 1 + 1  
16 ````  
17  
18 You can add options to executable code like this  
19  
20 ```{r}  
2:1 # My First Document
```

The RStudio interface includes the following panels:

- Source** panel: Shows the Quarto document code.
- Environment** panel: Shows the Global Environment with the message "Environment is empty".
- Console** panel: Shows the command being run: `==> quarto preview myFirstDoc.qmd --to html --no-watch-inputs --no-browse`. A red box highlights this area.
- Output** panel: Shows the processing details:

```
processing file: myFirstDoc.qmd  
output file: myFirstDoc.knit.md  
  
pandoc  
  to: html  
  output-file: myFirstDoc.html  
  standalone: true  
  section-divs: true  
  html-math-method: mathjax  
  wrap: none  
  default-image-extension: png
```



Take A Breath!

Rendering Quarto Docs

We will go over:

1. Rendering a PDF, HTML and Word document.
2. Common YAML options and add some to our document.
3. Different formatting options.
4. Inserting:
 1. Hyperlinks
 2. Figures
 3. Tables
5. R Code Chunks.

My First Quarto HTML

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:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

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

Customizing your quarto documents

We have options to customize our document

Quarto extensions is a powerful tool:

- Help modify and customize the defaults
- Creating shareable templates
- Adding additional functionalities to Quarto

Also, it is easy to get the extension you like!

```
quarto add extension <repo_name>  
quarto use template <repo_name>
```

<https://github.com/quarto-ext>

Quarto extension

Quarto extensions is a powerful tool:

- Help modify and customize the defaults
- Creating shareable templates
- Adding additional functionalities to Quarto

Also, it is easy to get the extension you like!

```
quarto add extension <repo_name>  
quarto use template <repo_name>
```

README.md

Extensions for Quarto

The `quarto-ext` organization collects extensions maintained by the core Quarto team.

Learn more about creating your extensions here: <https://quarto.org/docs/extensions/>

| Extension name | Description | Install |
|------------------------------------|---|---|
| latex-environment | Quarto extension to output custom LaTeX environments. | <code>quarto add extension quarto-ext/latex-environment</code> |
| lightbox | Create lightbox treatments for images in your HTML documents. | <code>quarto add extension quarto-ext/lightbox</code> |
| fontawesome | Use Font Awesome icons in HTML and PDF documents. | <code>quarto add extension quarto-ext/fontawesome</code> |
| pointer | A very simple RevealJS plugin extension that adds support for switching the cursor to a 'pointer' style element while presenting. | <code>quarto add extension quarto-ext/pointer</code> |
| attribution | Display attribution text sideways along the right edge of Revealjs slides. | <code>quarto add extension quarto-ext/attribution</code> |
| shinylive | Quarto extension to embed Shinylive for Python applications | <code>quarto add extension quarto-ext/shinylive</code> |
| fancy-text | Output nicely formatted versions of fancy strings such as LaTeX and BibTeX in multiple formats. | <code>quarto add extension quarto-ext/fancy-text</code> |
| holder | Quarto extension to create placeholder images | <code>quarto add extension quarto-ext/holder</code> |
| include-code-files | Quarto extension to include code from source files. | <code>quarto add extension quarto-ext/include-code-files</code> |

<https://github.com/quarto-ext>

Basic extension for IU*

<https://github.com/Mutaz94/quarto-CTSI-2024/>

The image shows a presentation slide with the following content:

- Title:** CTSI Slides Template
- Author:** First Last
- Date:** 2024-02-28
- Page Number:** 1 (partially visible at the bottom left)
- Page Footer:** CTSI 2024
- Logo:** Indiana CTSI Clinical and Translational Sciences Institute logo, featuring three overlapping hexagons in yellow, red, and blue.

*Unofficial/unapproved templates that **might or might not** follow some IU formatting rules

Improve your skills beyond this talk

Helpful References:

General resources:

- <https://quarto.org/docs/get-started/hello/rstudio.html>
- <https://www.youtube.com/watch?v=yvi5uXQMvu4>
- <https://www.youtube.com/watch?v=EbAAmrB0luA>
- <https://r4ds.hadley.nz/quarto.html>

YAML Header options

- <https://quarto.org/docs/reference/formats/opml.html>
- For HTML output: <https://quarto.org/docs/reference/formats/html.html>

Inserting figures

- <https://quarto.org/docs/authoring/figures.html>

Report and tips

- <https://github.com/Mutaz94/report>
- <https://github.com/Mutaz94/quarto-tips>

Take home messages

- Quarto is awesome!
- Reproducibility and maintainability
- Customizability
- Productivity



QUESTIONS?

