

Quarto

Elevating R Markdown for Advanced Publishing

Christophe Dervieux

Posit PBC

July 9, 2024

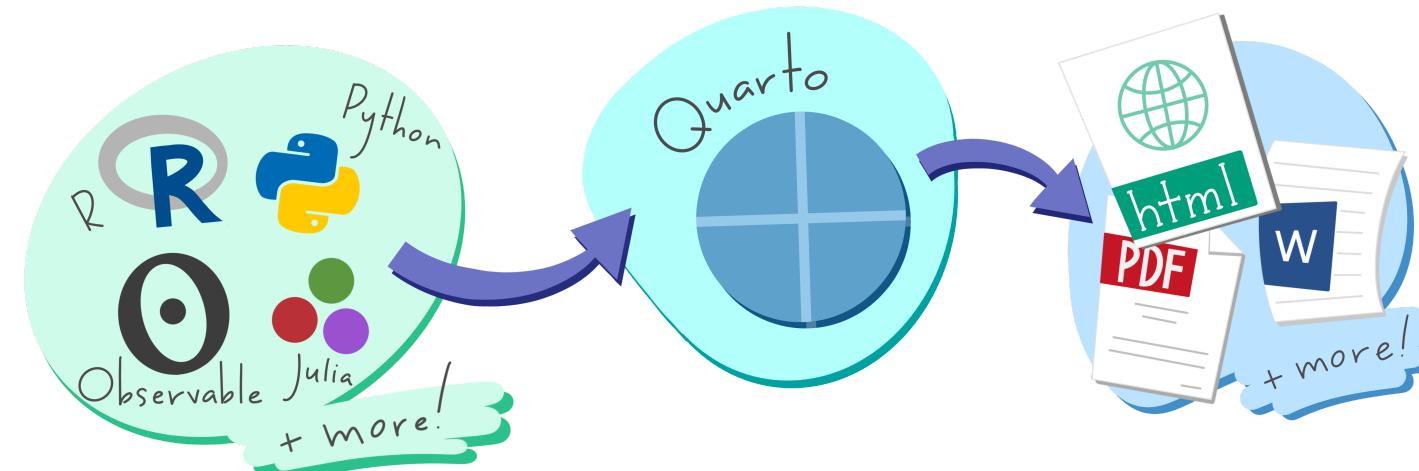


What is Quarto?

An open-source, scientific and technical publishing system

Quarto is the next generation of R Markdown

- it **unifies** and **improves** the R Markdown ecosystem
- it **extends** it for people who don't know R Markdown

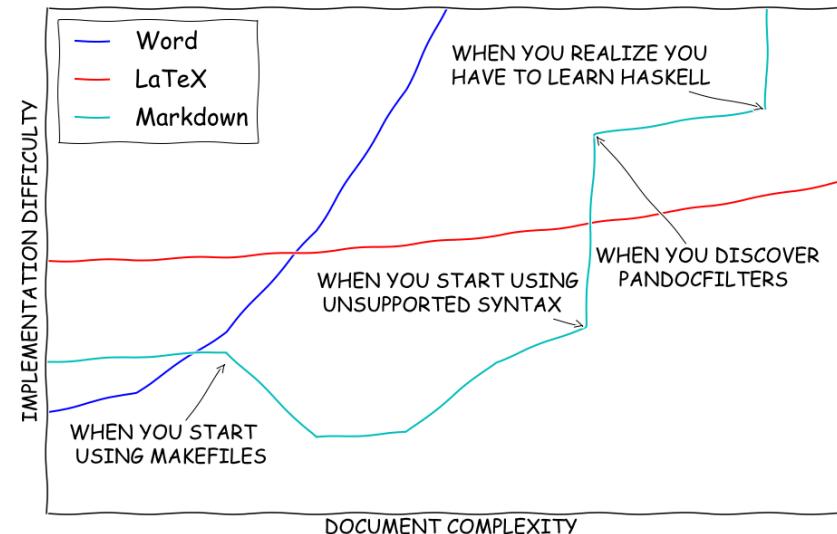


Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

Follow along at <https://cderv.github.io/user2024-quarto-talk/> and <https://quarto.org>

Why Quarto?

- Computational Documents and Scientific Markdown made easy for Single Source Publishing
- Designed from the start as multi-engine
- An attempt to bring +10 years experience of R Markdown to everyone!



Follow along at <https://cderv.github.io/user2024-quarto-talk/> and <https://quarto.org>

How to use Quarto?

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 $ quarto --help
2
3 Usage:   quarto
4 Version: 1.5.53
5
6 Description:
7
8   Quarto CLI
9
10 Options:
11
12   -h, --help      - Show this help.
13   -V, --version   - Show the version number for this program.
14
15 Commands:
```

How to use Quarto?

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 $ quarto --help
2
3 Usage:   quarto
4 Version: 1.5.53
5
6 Description:
7
8   Quarto CLI
9
10 Options:
11
12   -h, --help      - Show this help.
13   -V, --version   - Show the version number for this program.
14
15 Commands:
```

How to use Quarto?

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 $ quarto --help
2
3 Usage:   quarto
4 Version: 1.5.53
5
6 Description:
7
8   Quarto CLI
9
10 Options:
11
12   -h, --help      - Show this help.
13   -V, --version   - Show the version number for this program.
14
15 Commands:
```

How to use Quarto?

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.

```
10 Options:  
11  
12 -h, --help      - Show this help.  
13 -V, --version   - Show the version number for this program.  
14  
15 Commands:  
16  
17 render    [input] [args...]      - Render files or projects to various document types.  
18 preview   [file] [args...]       - Render and preview a document or website project.  
19 serve     [input]                - Serve a Shiny interactive document.  
20  
21 create    [type] [commands...]  - Create a Quarto project or extension  
22  
23 use       <type> [target]      - Automate document or project setup tasks.  
24 add       <extension>          - Add an extension to this folder or project
```

How to use Quarto?

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.

```
11
12   -h, --help      - Show this help.
13   -V, --version   - Show the version number for this program.
14
15 Commands:
16
17   render      [input] [args...]      - Render files or projects to various document types.
18   preview     [file]  [args...]      - Render and preview a document or website project.
19   serve       [input]                - Serve a Shiny interactive document.
20
21   create      [type]  [commands...] - Create a Quarto project or extension
22
23   use        <type> [target]      - Automate document or project setup tasks.
24   add        <extension>        - Add an extension to this folder or project
25   update     [target...]        - Updates an extension or global dependency.
```

How to use Quarto?

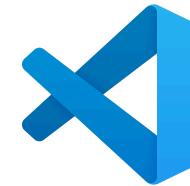
Quarto is a **command line interface (CLI)** that renders plain text formats (`.qmd`, `.rmd`, `.md`) OR mixed formats (`.ipynb`/Jupyter notebook) into static PDF/Word/HTML reports, books, websites, presentations and more.

17	render	<code>[input] [args...]</code>	- Render files or projects to various document types.
18	preview	<code>[file] [args...]</code>	- Render and preview a document or website project.
19	serve	<code>[input]</code>	- Serve a Shiny interactive document.
20			
21	create	<code>[type] [commands...]</code>	- Create a Quarto project or extension
22			
23	use	<code><type> [target]</code>	- Automate document or project setup tasks.
24	add	<code><extension></code>	- Add an extension to this folder or project
25	update	<code>[target...]</code>	- Updates an extension or global dependency.
26	remove	<code>[target...]</code>	- Removes an extension.
27			
28	convert	<code><input></code>	- Convert documents to alternate representations.
29	pandoc	<code>[args...]</code>	- Run the version of Pandoc embedded within Quarto.
30	typst	<code>[args...]</code>	- Run the version of Typst embedded within Quarto.
31	run	<code>[script] [args...]</code>	- Run a TypeScript, R, Python, or Lua script.

How to use Quarto?

Quarto integrates with other tools.

How to use Quarto?



Quarto integrates with other tools.

The screenshot shows the Quarto Web interface with the following components:

- EXPLORER**: A sidebar on the left listing files and folders under 'QUARTO-WEB' and 'docs'.
- diagrams.qmd x**: The main editor pane showing the content of the 'diagrams.qmd' file. The code includes Mermaid syntax for a flowchart and a note about the underlying language of Graphviz.
- Render**: A dropdown menu in the top right of the editor.
- Quarto Preview x**: A preview pane on the right showing the rendered output. It features a 'quarto' logo, a title 'Diagrams', and an 'Overview' section. Below it is a code block with Mermaid syntax, followed by a rendered flowchart diagram.

Code Content (diagrams.qmd):

```
## Overview

Quarto has native support for embedding [Mermaid](https://mermaid-js.github.io/mermaid/#/) and [Graphviz](https://graphviz.org/) diagrams. This enables you to create flowcharts, sequence diagrams, state diagrams, gnatt charts, and more using a plain text syntax inspired by markdown.

For example, here we embed a flowchart created using Mermaid:

````mermaid
graph LR
 A[Hard edge] --> B(Round edge)
 B --> C{Decision}
 C --> D[Result one]
 C --> E[Result two]
````

As illustrated above, diagrams are embedded using `mermaid` and `dot` executable cells (`dot` is underlying language of Graphviz)

Diagrams are a recent addition to Quarto---you should make sure to update to the [very latest](/docs/get-started/) version of Quarto (v0.9.518 or higher) before trying out the features described below.

::: callout-note
For print output formats like `pdf` or `docx`, diagram rendering makes use of the Chrome or Edge web browser to create a high-quality PNG. Quarto can automatically use an existing version of Chrome or Edge on your system, or alternatively if you don't have either installed, can use a lighter-weight library version of Chrome (see [Chrome Install](#chrome-install) below for details).
:::

## Mermaid

Mermaid is a Javascript based diagramming and charting tool that uses Markdown-inspired text definitions and a renderer to create and modify complex diagrams.
```

Preview Output:

Diagrams

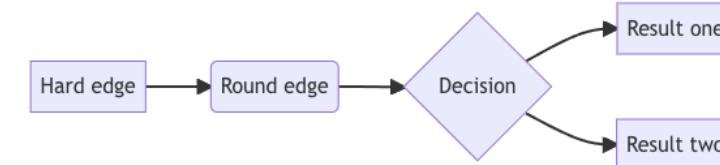
Overview

Quarto has native support for embedding [Mermaid](#) and [Graphviz](#) diagrams. This enables you to create flowcharts, sequence diagrams, state diagrams, gnatt charts, and more using a plain text syntax inspired by markdown.

For example, here we embed a flowchart created using Mermaid:

```
```mermaid
graph LR
 A[Hard edge] --> B(Round edge)
 B --> C{Decision}
 C --> D[Result one]
 C --> E[Result two]
````
```

```
graph LR
    A[Hard edge] --> B(Round edge)
    B --> C{Decision}
    C --> D[Result one]
    C --> E[Result two]
```



How to use Quarto?

Quarto integrates with other tools.



The screenshot shows a JupyterLab interface. On the left, the Explorer sidebar shows a file named "quarto-jupyterlab.ipynb". The main area displays a Quarto configuration file (qmd) with the following content:

```
---
```

```
title: Matplotlib Demo
author: Norah Smith
date: 'May 22nd, 2021'
format:
  html:
    code-fold: true
keep-ipynb: true
---
```

Polar Axis

For a demonstration of a line plot on a polar axis, see [@fig-polar](#).

In cell [2], the code for generating a polar plot is shown:

```
[2]: #| 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'})
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```

The resulting polar plot is displayed at the bottom of the cell.

The screenshot shows a Quarto preview window titled "Matplotlib Demo" with the author "Norah Smith" and date "May 22nd, 2021". The section "Polar Axis" contains the following text:

For a demonstration of a line plot on a polar axis, see [fig. 1](#).

A "Code" button is present. Below it is a polar plot with concentric circles and a blue line plot forming a spiral pattern.

How to use Quarto?

Quarto integrates with other tools.



diagrams.qmd — quarto-web

EXPLORER Render ... Quarto Preview ... quarto-tutorials — ~ ...

hello.qmd Go to file/function Addins ...

Source Visual ABC Render Run ...

Format Insert Table Outline

```
---
```

```
title: "Hello, Quarto"
format: html
editor: visual
---
```

```
{r}
#| label: load-packages
#| include: false

library(tidyverse)
library(palmerpenguins)
```

Meet Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Meet the penguins

The `penguins` data from the `palmerpenguins` package contains size measurements for `nrow(penguins)` penguins from three species observed on three islands in the Palmer Archipelago, Antarctica.

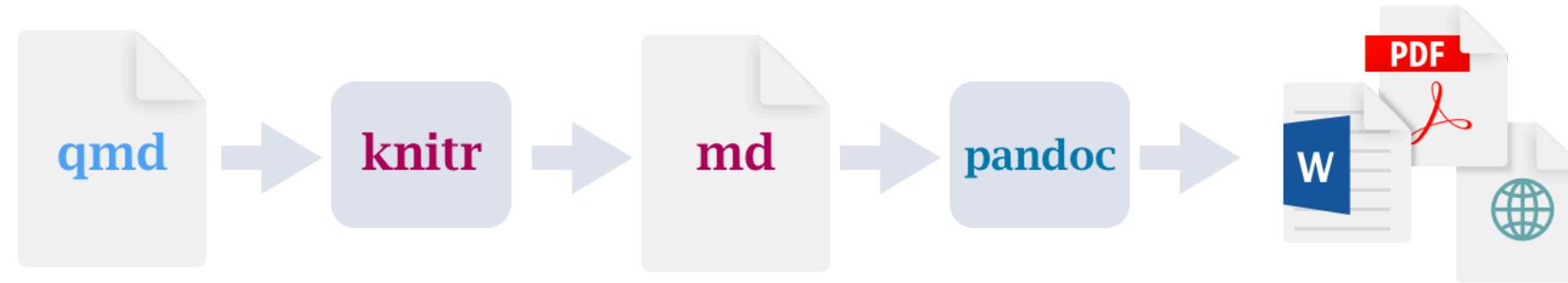
CHINSTRAP! GENTOO! ADÉLIE!

The plot below shows the relationship between flipper and bill lengths of these penguins.

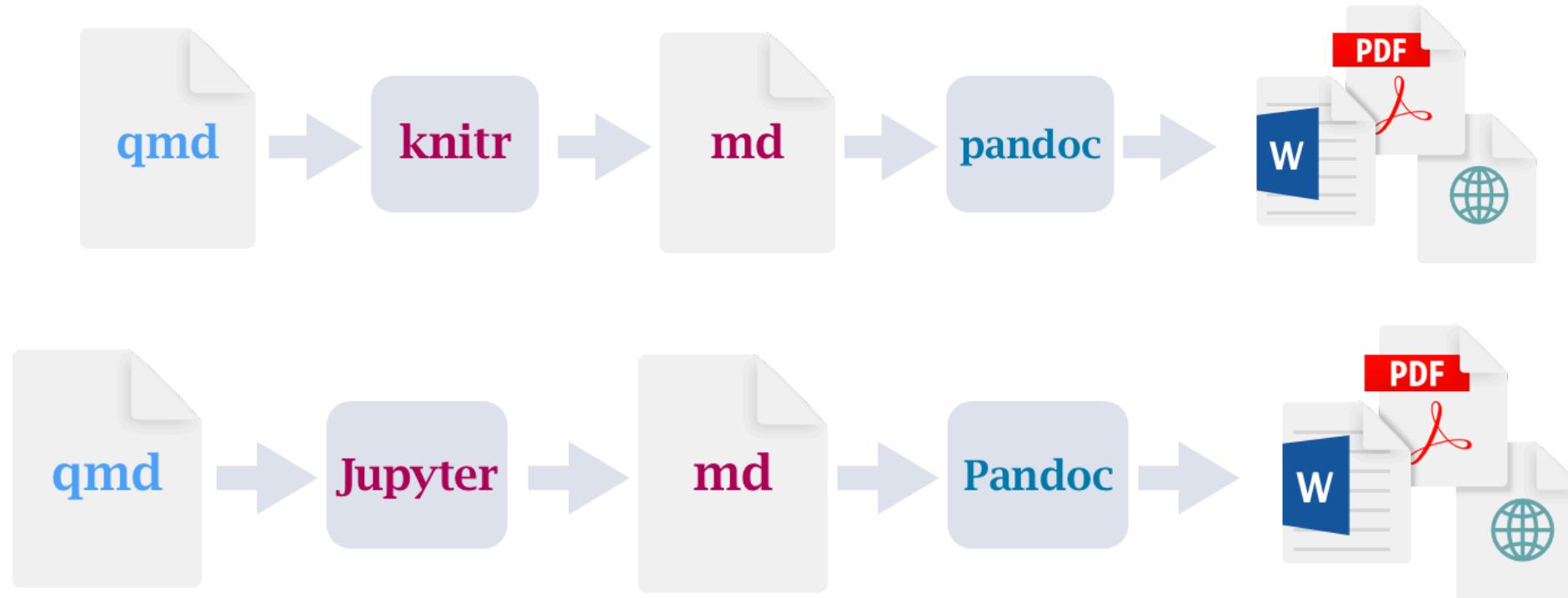
Flipper and bill length



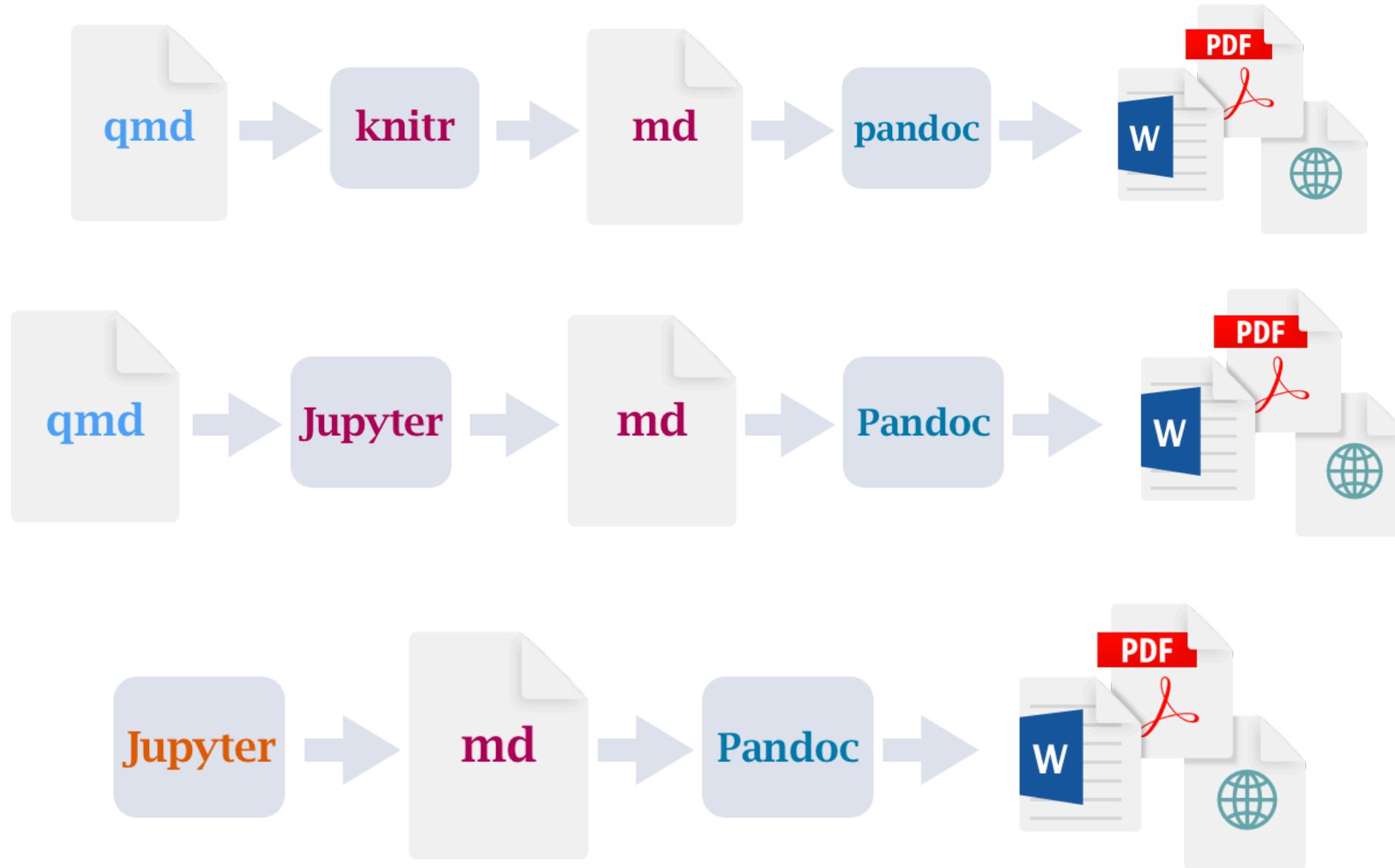
Quarto Workflow



Quarto Workflow



Quarto Workflow



Quarto Syntax

From R Markdown...

```
1 ---  
2 title: Hello penguins!  
3 output:  
4   html_document:  
5     toc: true  
6 ---  
7  
8 Some **Markdown** content based on _Pandoc_ 's markdown  
9  
10 ````{r species, fig.width=5, fig.cap="About penguins"}  
11 ggplot(penguins) + aes(x = bill_length_mm, y = bill_depth_mm, color = species) +  
12   geom_point() + labs(x = "Bill length (mm)", y = "Bill depth (mm)")  
13 ````
```



Quarto Syntax

From R Markdown...

```
1 ---  
2 title: Hello penguins!  
3 output:  
4   html_document:  
5     toc: true  
6 ---  
7  
8 Some **Markdown** content based on _Pandoc_ 's markdown  
9  
10 ````{r species, fig.width=5, fig.cap="About penguins"}  
11 ggplot(penguins) + aes(x = bill_length_mm, y = bill_depth_mm, color = species) +  
12   geom_point() + labs(x = "Bill length (mm)", y = "Bill depth (mm)")  
13 ````
```



Quarto Syntax

... to Quarto

```
1 ---  
2 title: Hello penguins!  
3 format: html  
4 toc: true  
5 ---  
6  
7 Some **Markdown** content based on _Pandoc_ 's markdown  
8  
9 ````{r}  
10 #| label: species  
11 #| fig-width: 5  
12 #| fig-cap: About penguins  
13 ggplot(penguins) + aes(x = bill_length_mm, y = bill_depth_mm, color = species) +  
14   geom_point() + labs(x = "Bill length (mm)", y = "Bill depth (mm)")  
15 ````
```



Quarto Syntax

... to Quarto

```
1 ---  
2 title: Hello penguins!  
3 format: html  
4 toc: true  
5 ---  
6  
7 Some **Markdown** content based on _Pandoc_ 's markdown  
8  
9 ````{r}  
10 #| label: species  
11 #| fig-width: 5  
12 #| fig-cap: About penguins  
13 ggplot(penguins) + aes(x = bill_length_mm, y = bill_depth_mm, color = species) +  
14   geom_point() + labs(x = "Bill length (mm)", y = "Bill depth (mm)")  
15 ````
```



Quarto Syntax

... to Quarto

```
1 ---  
2 title: Hello penguins!  
3 format: html  
4 toc: true  
5 ---  
6  
7 Some **Markdown** content based on _Pandoc_ 's markdown  
8  
9 ````{r}  
10 #| label: species  
11 #| fig-width: 5  
12 #| fig-cap: About penguins  
13 ggplot(penguins) + aes(x = bill_length_mm, y = bill_depth_mm, color = species) +  
14   geom_point() + labs(x = "Bill length (mm)", y = "Bill depth (mm)")  
15 ````
```



Quarto formats

Multiple fresh polished formats, built-in, with consistent syntax and features.

| Feature | Quarto | R Markdown |
|---------------|----------|-------------------------|
| Basic Formats | html | html_document |
| | pdf | pdf_document |
| | docx | word_document |
| | typst | |
| Beamer | beamer | beamer_presentation |
| PowerPoint | pptx | powerpoint_presentation |
| HTML Slides | revealjs | xaringan |
| | | ioslides |
| | | revealjs |

Quarto formats

Multiple fresh polished formats, [built-in](#), with consistent syntax and features.

| Feature | Quarto | R Markdown |
|------------------|---|---|
| Cross References | Quarto Crossrefs | html_document2
pdf_document2
word_document2 |
| Advanced Layout | Quarto Article Layout | tufte
distill |
| Websites & Blogs | Quarto Websites
Quarto Blogs | blogdown
distill |
| Books | Quarto Books | bookdown |
| Interactivity | Quarto Interactive Documents | Shiny Documents |
| Dashboards | Quarto Dashboards | flexdashboard |

Focus: HTML Theming

Quarto theme files are based on SCSS for all HTML output formats (shared by HTML and RevealJS slides, extendable to work with anything).

For bootstrap document, based on core variables defined by `bslib` (so fully compatible with Shiny, Flexdashboard, & R Markdown themes).

`my-style.scss` (Quarto theme file)

```
/*-- scss:defaults --*/
$h2-font-size:         1.6rem !default;
$headings-font-weight: 500 !default;

$custom-color: blue;

/*-- scss:rules --*/
h1, h2, h3, h4, h5, h6 {
  text-shadow: -1px -1px 0 rgba(0, 0, 0, .3);
  color: $custom-color;
}
```

`_quarto.yml` or YAML header

```
format:
  html:
    linkcolor: darkgrey
    theme: [litera, my-style.scss]
```

Focus: Typst CSS for nice table output in PDF

New PDF rendering through Typst (<https://typst.app>)

format: typst

This is the quickest way to do PDF with already nice features !

```
# Producing a HTML Table with CSS styles
library(gt)

temp <- data.frame(
  year = c(1920:1924),
  Jan = c(40.6, 44.2, 37.5, 41.8, 39.3),
  Jun = c(58.5, 58.7, 57.8, 52.7, 57.7)
)

nice_col <- scales::col_numeric(
  colorspace::diverge_hcl(
    n = 9, palette = "Green-Orange"
  ), domain = c(35, 62)
)

gt(temp) |>
  data_color(
    columns = c(-year),
    fn = nice_col
  )
```

| year | Jan | Jun |
|------|------|------|
| 1920 | 40.6 | 58.5 |
| 1921 | 44.2 | 58.7 |
| 1922 | 37.5 | 57.8 |
| 1923 | 41.8 | 52.7 |
| 1924 | 39.3 | 57.7 |

Rendered in PDF with Typst

Focus: Publishing

Integrated `quarto publish` command that targets a variety of services

```
1 > quarto publish --help
2
3 Usage: quarto publish [provider] [path]
4 Version: 1.5.52
5
6 Description:
7
8 Publish a document or project to a provider.
9
10 Available providers include:
11
12 - Quarto Pub (quarto-pub)
13 - GitHub Pages (gh-pages)
14 - Posit Connect (connect)
15 - Posit Cloud (posit-cloud)
16 - Netlify (netlify)
17 - Confluence (confluence)
18 - Hugging Face Spaces (huggingface)
19
20 Accounts are configured interactively during publishing.
21 Manage/remove accounts with: quarto publish accounts
```



Focus: Shortcodes

Quarto specific syntax to include content:

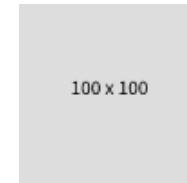
<https://quarto.org/docs/authoring/shortcodes.html>

Let's add some sample image !

```
![A caption]({{< placeholder >}})
```

placeholder is one of the built-in
shortcodes

Let's add some sample image !



A caption

Shortcodes is among what is extensible using Quarto Extensions

Quarto Projects

A Quarto Project is a directory that contains a file called `_quarto.yml`.

Organizing content as a project to gain project's features and enhanced outputs

```
# from project root  
quarto render
```

- Projects have a type (website, book, manuscript, ...) driving behaviors and specific features
- Shared metadata from project to sub-folder level
- Reproducibility with computations can be controlled (cache, freeze)
- Pre and Post render scripts can be added
- Project Profile for distinct project behavior



Quarto Projects

Examples of project's configurations

_quarto.yml (simple website)

```
project:  
  type: website  
  
website:  
  title: "today"  
  favicon: logo.png  
  navbar:  
    left:  
      - href: index.qmd  
        text: Home  
      - about.qmd  
  
format:  
  html:  
    theme: cosmo  
    css: styles.css  
    toc: true  
  
freeze: auto
```

_quarto.yml (simple book)

```
project:  
  type: book  
  
book:  
  title: "mybook"  
  author: "Jane Doe"  
  date: "8/18/2021"  
  chapters:  
    - index.qmd  
    - intro.qmd  
    - references.qmd  
  
bibliography: references.bib  
  
format:  
  html: default  
  pdf: default  
  epub:  
    cover-image: cover.png
```



Learning More

This presentation aimed to give you a taste of what Quarto can do and a broad overview of its features.

Follow other talks at useR! 2024 to learn more about Quarto, and our resources below.

Resources

- Getting started: <https://quarto.org/docs/get-started/>
- User guide: <https://quarto.org/docs/guide/>
- Awesome Quarto: <https://github.com/mcanouil/awesome-quarto>

Questions?

