Reproducible and Automated Reporting with Quarto

Pasky Miranda (UKRI Digital Research Skills Catalyst)

https://tinyurl.com/QuartoBES

This workshop has been developed by the UKRI Digital Research Skills Catalyst.

Housekeeping

This tutorial is available as a webpage here: github.
If you don't have R Studio installed or working, no problem! It's also available as a positcloud instance: positcloud.

If you see this emoji then it's an instruction to do something.

What You'll Learn Today

- Markdown basics
- How to create a Quarto document
- What a YAML does

- Adding content to a Quarto document
- How to render Inline code
- How to render Figures + tables
- How to use Auto-renumbering
- How to use Visual editor
- How to add Citations

What is Quarto?

Quarto is a modern scientific publishing system for **dynamic, reproducible documents**.

- Write text + code together
- Render to **HTML**, **PDF**, **Word**, **slides**, **manuscripts**, more
- Supports R, Python, Julia, Observable
- Like R Markdown, but more flexible + standardised

Why Use Quarto?

Typical write-up workflow:

- Figures/tables manually updated
- Graphs are saved to file then inserted manually
- Cross-references break when ordering changes
- Reformatting for journals takes hours
- Lots of copying/pasting
- Lots of potential for errors

With Quarto you don't have to worry about any of these. It does the work for you!

Publishing

To address the reproducibility crisis journals now increasingly want you to have your code/workbook submitted with the manuscript. Working in Quarto allows you have all of that all in one place, readily exportable to whichever format the journal requires.

Demo: Multiple Formats

These are all the **same** `.qmd` document, just with different output formats:

- HTML
- PDF
- Word document
- Revealjs slide deck

(Some tidying up may be required for certain formats, but it all can run from the same source document!)

Markdown Basics

- The YAML header sets the default behaviour for the document and is between `---` at the top of the document (more on this later)
- R Code chunks are between ```` ```{r} ```` and ```` ``` ```` and chunk options, starting `#|`, determine how/whether they run whether code/output is included in the rendered document
- You can run code chunk interactively or through rendering
- Comments: `|#`, `#` in code chunks, `<!-- in text -->` but use Ctrl+Shift+C

Text Formatting

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

Headings