Introduction to Quarto

A Modern Approach to Scientific Publishing

Cédric Grueau

cedric.grueau@isel.ips.pt

Instituto Superior de Engenharia de Lisboa

2025-10-23

Research Outputs

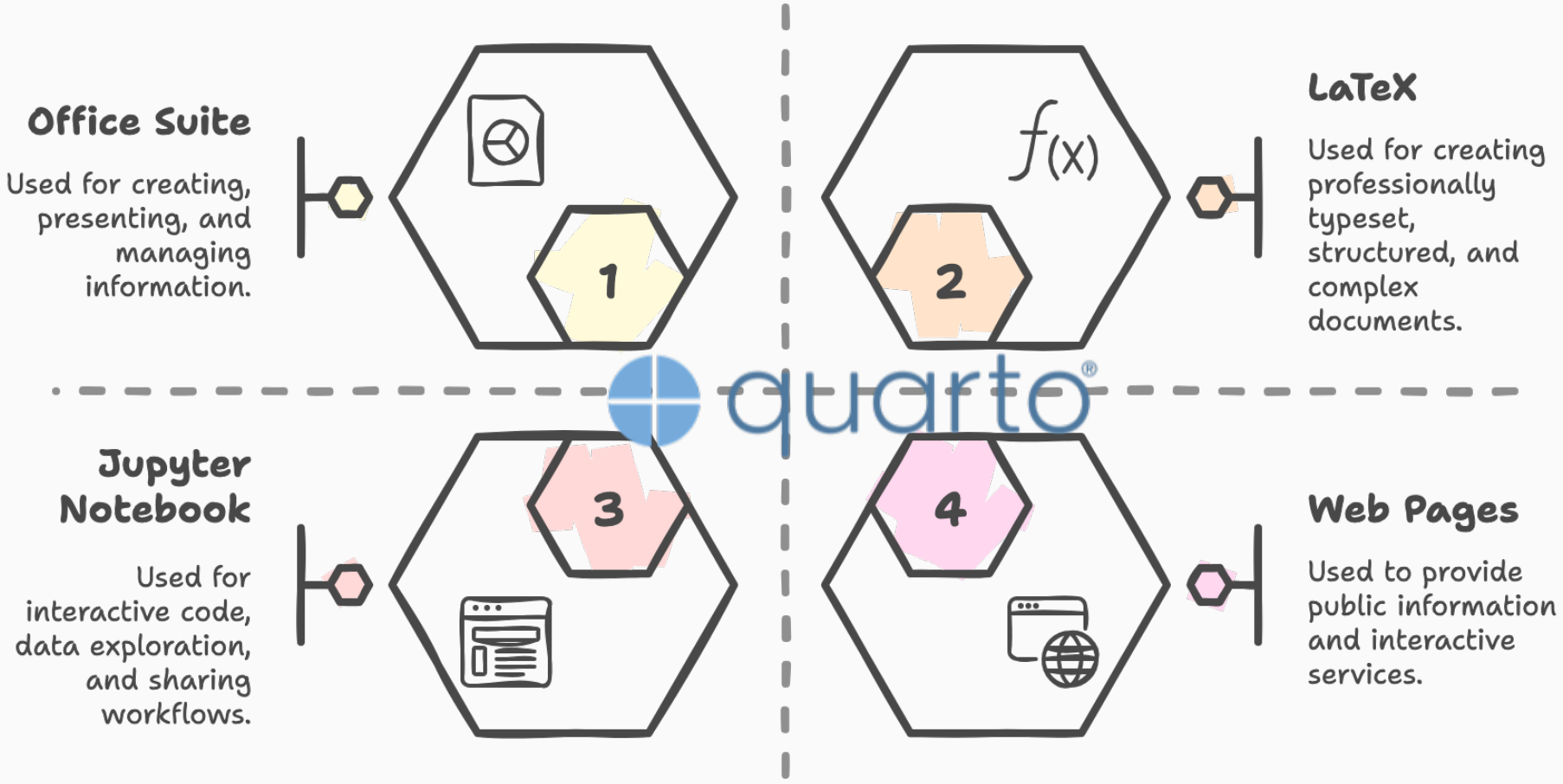
Among many others, here are some of the outputs created by researchers:

- Research Papers, Books, or Reports
- Research Presentations (Conference or other Stakeholders)
- Personal, Project, or Conference Websites

However, we are spending **far too much time working on the content AND on the design** of our communications.

> And if you teach, you can add:
>
> - Lecture notes, Presentations, Assignments, Exam questions, ...

What is Quarto?



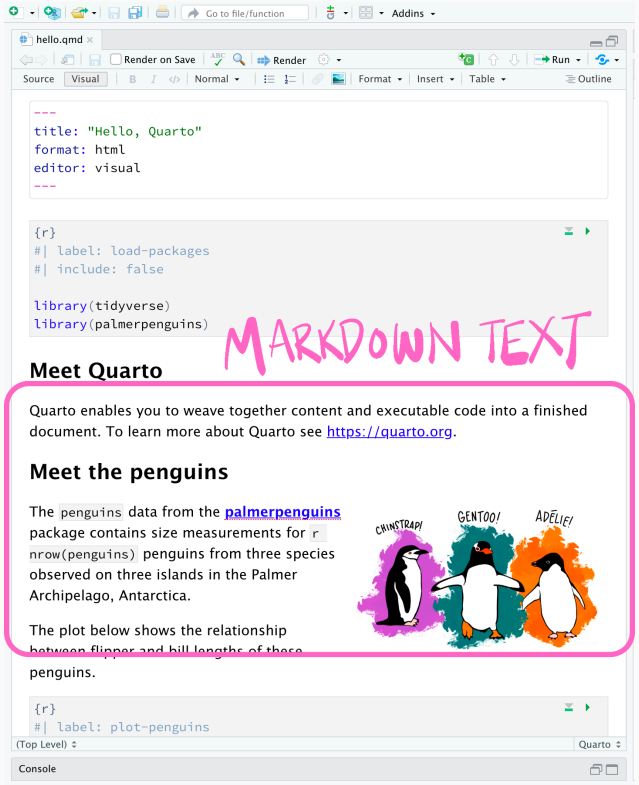
What is Quarto?

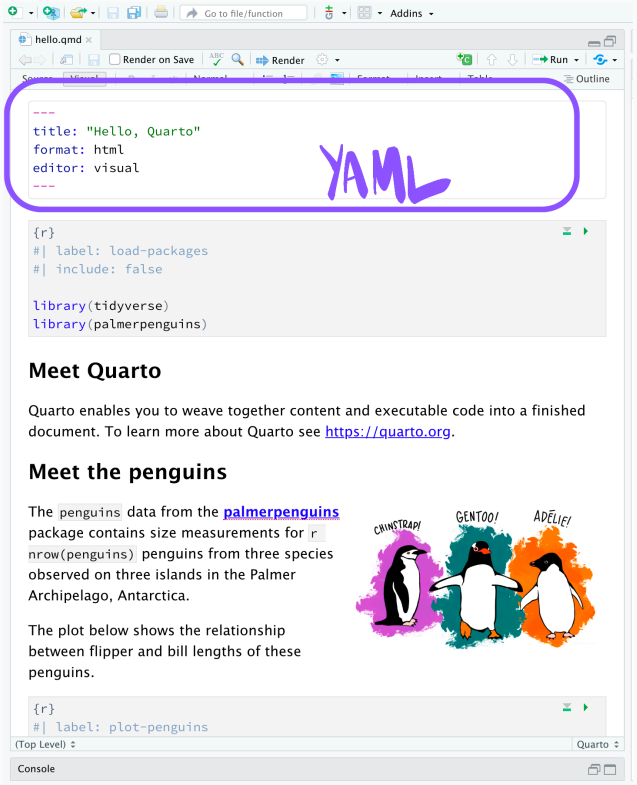
Quarto is an **open-source scientific and technical publishing system** that builds on standard markdown with features essential for scientific communication, creating rich, interactive, and computational documents.

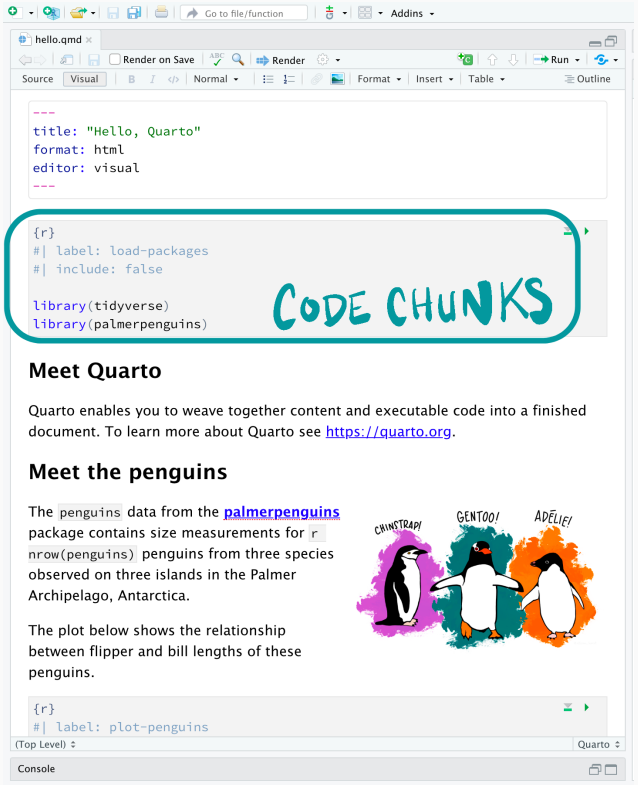
In simple terms

- It is command-line software that creates slick and polished documents.
- It uses plain text but can output PDF, Word, PowerPoint, or HTML files from a single source.
- It plugs into most coding editors like VS Code, Jupyter, RStudio and more but code is not necessary to make it work.

How does it Work?



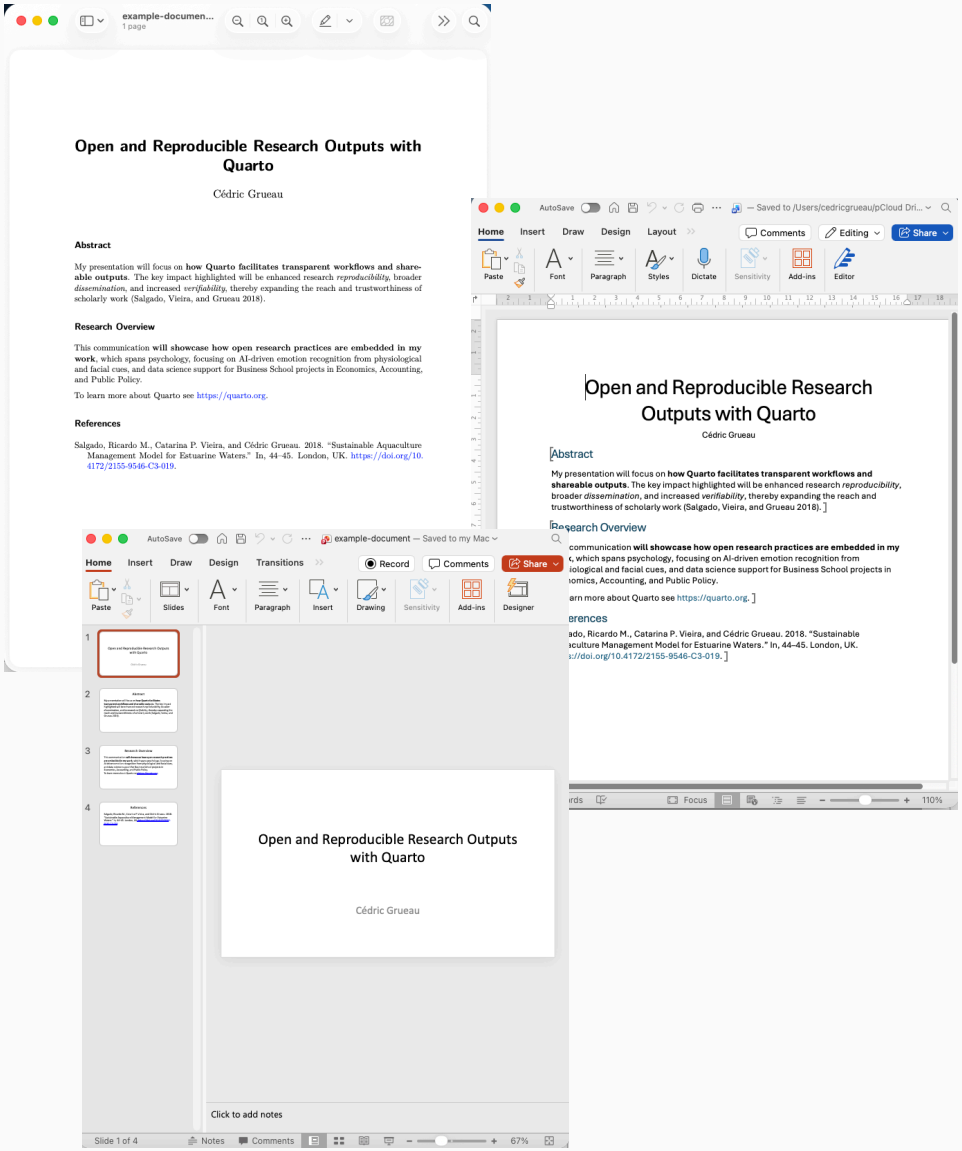




Simple Example

my_first_quarto.qmd

```yaml
---
title: Open and Reproducible Research Outputs with Quarto
author: Cédric Grueau
format:
  pdf: default
  docx: default
  pptx: default

bibliography: bibliography.bib
---

## Abstract

My presentation will focus on **how Quarto
facilitates transparent workflows and shareable
outputs**. The key impact highlighted will be enhanced
research *reproducibility*, broader *dissemination*, and
increased *verifiability*, thereby expanding the reach
and trustworthiness of scholarly work [@dupre2025open].

## Research Overview

This communication **will showcase how open research
practices are embedded in my work**, which spans psychology,
focusing on AI-driven emotion recognition from physiological
and facial cues, and data science support for Business School
projects in Economics, Accounting, and Public Policy.

To learn more about Quarto see <https://quarto.org>.

## References
```



Quarto in my Research

Research Papers

```yaml
---
title: "Transforming Hiking Experiences: A Digital Platform for Sustainable Trail Management and Co
author:
  - name: Cédric Grueau
    orcid: 0000-0003-3445-4070
    email: cedric.grueau@isel.ips.pt
    affiliations:
      - id: uni
        name: Lisbon School of Engineering
        department: Departamento de Engenharia Informática
        address: Rua Conselheiro Emídio Navarro, 1
        city: Lisboa, Portugal
        postal-code: 1959-007
    attributes:
      corresponding: true

abstract: |
  Mountain tourism in Portugal suffers from fragmented digital infrastructure and trail data, limit
local landscapes and communities. This project addressed the dual challenge of data interoperabilit
the objective of transforming hiking into a driver of sustainable and inclusive mountain developmer
and sports science alongside key stakeholders: the Portuguese Federation of Sports for Disabilities
intervention consisted of an integrated platform featuring a standards-compliant (OGC, ISO 3021:202
profiles, and a location-based storytelling system. The approach was assessed through iterative pro
pilot case study in the Beijames Valley. The project delivered a fully functional platform that dem
accessibility information, empowering users with mobility limitations and raising awareness among a
awareness, while the robust backend established a foundation for data interoperability previously a
of overall data quality and user experience. Co-production with diverse stakeholders is essential f
moving beyond academic prototyping; we recommend establishing clear governance models and securing
evolve into lasting community resources.
keywords:
  - Mountain Tourism
  - Digital Platform
  - Co-design
  - Storytelling
  - Sustainable Development
  - Serra da Estrela
  - Portugal
date: last-modified
bibliography: bibliography.bib
format:
  elsevier-pdf:
    keep-tex: true
```

Transforming Hiking Experiences: A Digital Platform for Sustainable Trail Management and Community Engagement in Serra da Estrela, Portugal

Cédric Grueau[a,*]

[a]Lisbon School of Engineering, Departamento de Engenharia Informática, Rua Conselheiro Emídio Navarro, 1, Lisboa, Portugal, 1959-007

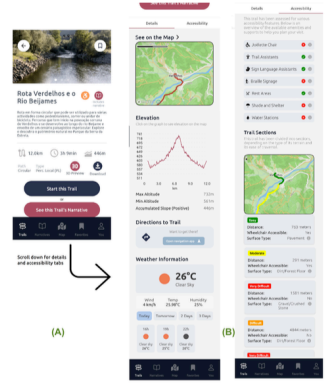
Abstract

Mountain tourism in Portugal suffers from fragmented digital infrastructure and trail data, limiting accessibility for diverse users and failing to foster deep, sustainable connections with local landscapes and communities. This project addressed the dual challenge of data interoperability and social exclusion by developing a collaborative digital platform, Tell-me a Trail, with the objective of transforming hiking into a driver of sustainable and inclusive mountain development. We employed a transdisciplinary, co-design methodology, engaging researchers in geomatics and sports science alongside key stakeholders: the Portuguese Federation of Sports for Disabilities (FPDD), the AllaBoard association, and the Friends of Serra da Estrela Association (ASE). The intervention consisted of an integrated platform featuring a standards-compliant (OGC, ISO 3021:2023) geospatial data framework, an inclusive mobile application with adaptive accessibility profiles, and a location-based storytelling system. The approach was assessed through iterative prototyping and user testing, including usability evaluations (System Usability Scale) and a pilot case study in the Beijames Valley. The project delivered a fully functional platform that demonstrated high usability. It successfully provided granular, section-by-section trail accessibility information, empowering users with mobility limitations and raising awareness among all hikers. The integration of co-produced narratives enhanced ecological and cultural awareness, while the robust backend established a foundation for data interoperability previously absent in Portugal. The primary lesson is that inclusive design is not a feature but a driver of overall data quality and user experience. Co-production with diverse stakeholders is essential for authenticity and addressing real-world needs. However, long-term sustainability requires moving beyond academic prototyping; we recommend establishing clear governance models and securing dedicated funding for content maintenance and technical

ager, allows administrators and local stakeholders to co-create and manage this content, which is delivered as text, images, audio, or video.

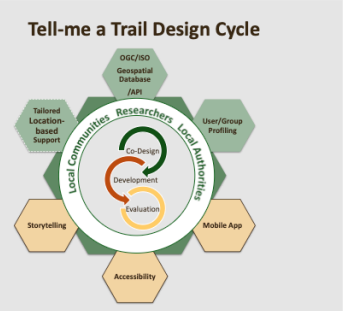
To ensure functionality in remote mountain environments with limited connectivity, the application features a robust offline mode. Users can download trails and their associated narratives beforehand, enabling full access to navigation and storytelling features during the activity.

The interface was refined through a multi-phase, iterative evaluation process. Screenshots from the mobile application (Figure 3) demonstrate its core functionality for inclusive planning, showing the trail selection screen filtered by user-defined accessibility profiles and the detailed, segmented trail view.



specialize in outdoor activities in mountain regions. This partnership marked a critical shift in the project's trajectory, explicitly orienting it towards principles of universal design.

To ensure these principles were authentically applied, the project engaged two key stakeholder organizations dedicated to accessibility: - The Portuguese Federation of Sports for People with Disabilities (FPDD) provided essential expertise on accessibility guidelines and the specific needs of users with reduced mobility. - The AllaBoard Association (based in Setubal, a southern city integrated in Arrábida Natural Park), with its experience in creating inclusive outdoor experiences for people on the autism spectrum and those with limited mobility or stamina, contributed vital insights into designing for a broad spectrum of cognitive and physical needs. - Finally, the ASE provided the crucial local context and territorial knowledge, ensuring that the platform's implementation was aligned with local conservation efforts and sustainable development goals and connecting the digital tool to the physical and cultural landscape it represents. This consortium—spanning geomatics, sports science, disability advocacy, and local environmental stewardship—formed the essential collaborative network that guided the co-design process from its inception.



esign network for the Tell-me a Trail platform

Presentations

Web GIS

Aplicações Web com características de Sistemas de Informação Geográfica... e IoT

Cédric Grueau

Introdução a cartografia para a Web com o quarto

Cédric Grueau

<MEEC 2023-24>

Websites

Introduction to Quarto

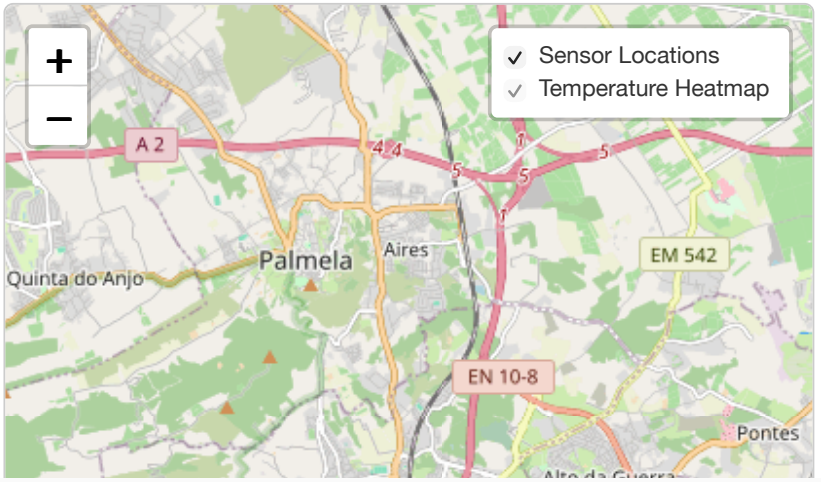
Introduction to Quarto

A Modern Approach to Scientific Publishing

Welcome to the Quarto Seminar

In the evolving landscape of academic publishing, the demand for reproducible research and dynamic content has never been greater. Traditional publishing workflows often create silos between data

Campus Sensor Network Visualization



Criar mapas para a Web com `quarto {ojs}` e `geoviz`

O objetivo desta sessão é introduzir o participante à criação de mapas interativos utilizando o **Quarto** e a biblioteca JavaScript **geoviz**.

1 Introdução

Antes de começar, por favor, certifique-se que leu a apresentação de introdução à Web, à sua história, às suas linguagens e ao Observable JavaScript. Ela é um pré-requisito

Scientific Curriculum

What else can we do with Quarto?

- Dashboards
  - https://mine.quarto.pub/olympic-games-py/
- DataScience notebooks
- Books
  - https://wesmckinney.com/book/
- University Thesis
  - https://github.com/Jupyter4Science/awesome-quarto-thesis

And for professores

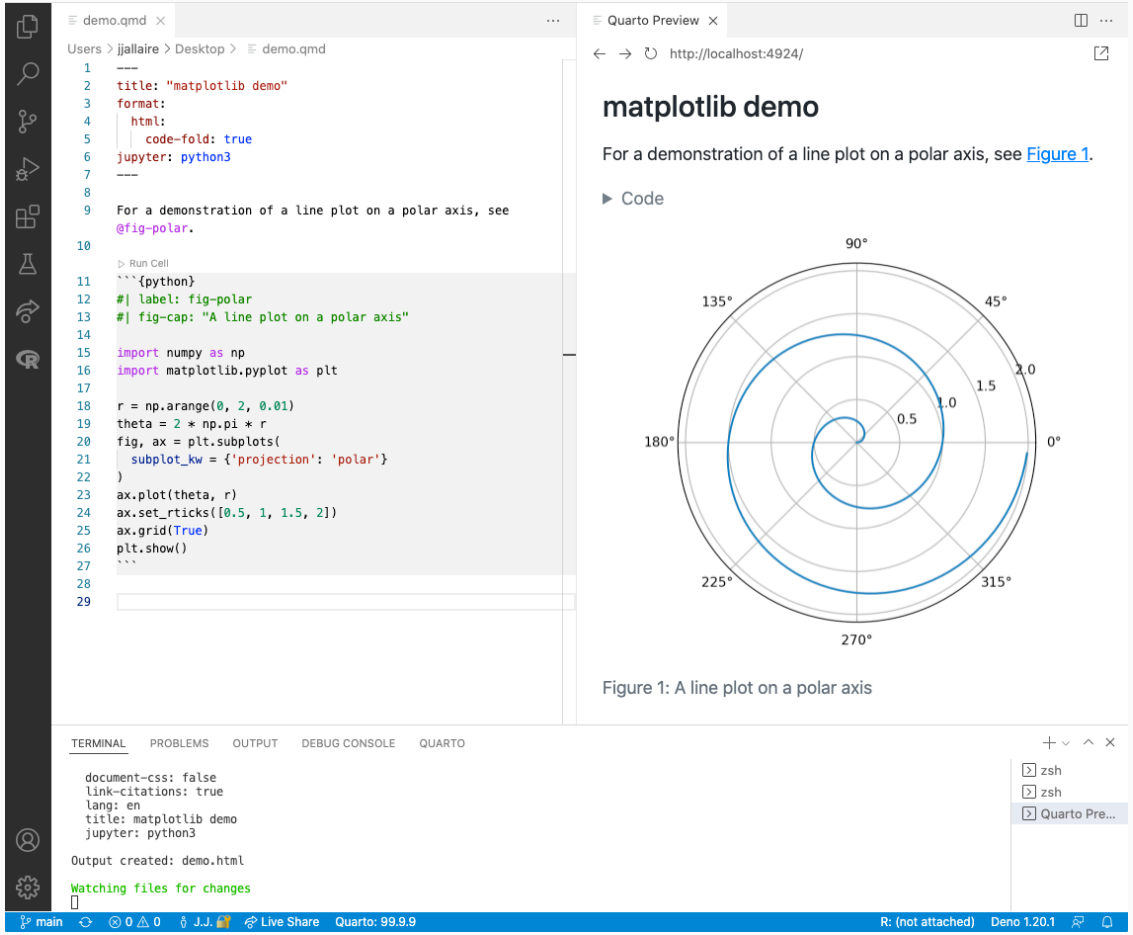
Lecture notes, exams, slides, assignments, ...

Key Use Cases for YOUR Projects

- **The Interactive Portfolio Thesis:** Your thesis website *is* your portfolio.
- **Computational Art & Design:** Show the code and the output side-by-side.
- **User Study Analysis:** Embed interactive dashboards of your results.
- **Prototype Documentation:** Embed live prototypes (WebGL, JS) directly into the documentation.

Quarto

Quarto Requirements

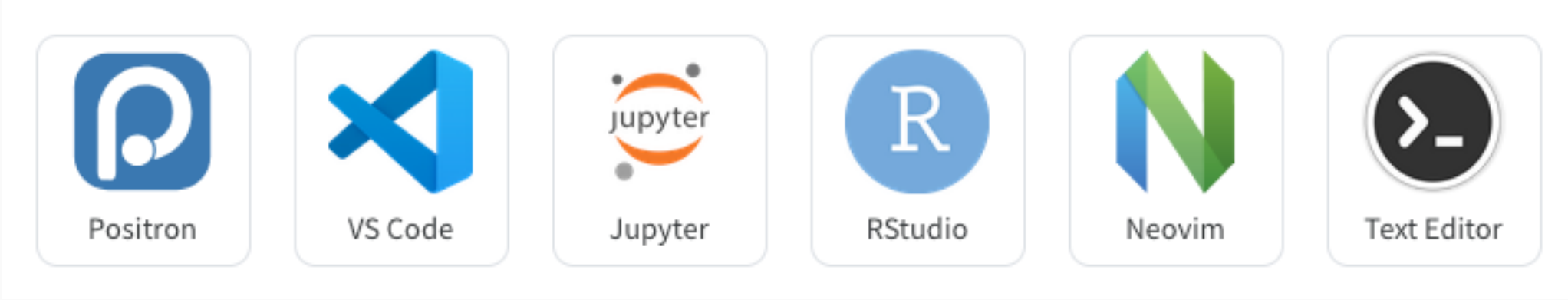


IDE (Text Editor) + Quarto + Latex (or Typst) + Programming Language

Quarto platforms

Quarto can be used with any platform.

A tutorial is offered for the following ones at https://quarto.org/docs/get-started/hello:



Quarto Structure

Quarto files have 3 different types of content:

1. The YAML

In a cell of type “Raw”, displayed between two series of `---` signs, it corresponds to the metadata shown in the header of the output file (e.g., title, author, date, ...) and the type of output (e.g., pdf, html, doc, ...)

2. The Text

Written in **Markdown style** (i.e., text without formatting), it is used as core description in the output document

3. The Code

Inserted in the Quarto inside code cells, the code is processed when creating the output and can display figures and tables

1. The YAML

The YAML

Simple

```
---
format: html
---
```

Default

```
---
title: Quarto Basics
format: html
date: "99/99/9999"
---
```

> ⚠ **Warning**
>
> indentation is very important, every line finishing with `:` involves **1 Tab** indentation on the following line.

```
---
title: Quarto Basics
date: "99/99/9999"
format:
  html:
    code-fold: true
---
```

2 - Markdown Style

Overview

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 Markdown for more in-depth documentation.

Text Formatting

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

Headings

Markdown Syntax	Output
`# Header 1`	Header 1
`## Header 2`	Header 2
`### Header 3`	Header 3
`#### Header 4`	Header 4
`##### Header 5`	Header 5
`###### Header 6`	Header 6

Links

Markdown Syntax	Output
`<https://quarto.org>`	https://quarto.org
`[Quarto](https://quarto.org)`	Quarto

Images

Images are created using a similar notation to links.

They are done in the format

```
1 ![]()
```

- The `!` lets markdown know it's an image
- Inside the `[]` you can optionally put a caption.
- Inside the `()` you place a link to the image relative to the location of the document OR on the web.

Web Image Example

`![]`

`(https://avatars.githubusercontent.com/u/107476423?s=400&u=84c1b49966ea958785a8954726c116b7952b2041&v=4)`

Local Image Example

```
![Image of a white robot surrounded by dashboards, holding a magnifying glass and thinking]
(resources/cover_image.jpeg)
```

This points towards an image where there is a folder called ‘resources’ at the same level as the .qmd file being written. It moves into the resources folder, then looks for an image called `cover_image.jpeg`.

Image of a white robot surrounded by dashboards, holding a magnifying glass and thinking

Images

Markdown Syntax	Output
`1 ![](image.png)`	
`1 ![Caption](image.png)`	 Caption
`1 [![Caption](image.png)](https://quarto.org)`	

List

```
* unordered list
    + sub-item 1
    + sub-item 2
        - sub-sub-item 1
```

- unordered list
  - sub-item 1
  - sub-item 2
    - sub-sub-item 1

```
*   item 2

    Continued (indent 4 spaces)
```

- item 2

  Continued (indent 4 spaces)

List

```
1. ordered list
2. item 2
    i) sub-item 1
        A.  sub-sub-item 1
```

1. ordered list
2. item 2
    i. sub-item 1
        a. sub-sub-item 1

```
- [ ] Task 1
- [x] Task 2
```

- ☐ Task 1
- ☑ Task 2

List

```
(@)  A list whose numbering

continues after

(@)  an interruption
```

1. A list whose numbering

   continues after

2. an interruption

```
::: {}
1. A list
:::

::: {}
1. Followed by another list
:::
```

1. A list

1. Followed by another list

```
term
: definition
```

term

: definition

List

Note

Note that unlike other Markdown renderers (notably Jupyter and GitHub), lists in Quarto require an entire blank line above the list.

Otherwise the list will not be rendered in list form, rather it will all appear as normal text along a single line.

```
Some text:

* First bullet point
* Second bullet point
```

Some text:

- First bullet point
- Second bullet point

```
Some text:
* First bullet point
* Second bullet point
```

Some text: * First bullet point * Second bullet point

Footnotes

Markdown supports numbering and formatting footnotes using the following syntax:

```
Here is a footnote reference,[^1] and another.[^longnote]

[^1]: Here is the footnote.

[^longnote]: Here's one with multiple blocks.

    Subsequent paragraphs are indented to show that they
belong to the previous footnote.

        { some.code }

    The whole paragraph can be indented, or just the first
    line.  In this way, multi-paragraph footnotes work like
    multi-paragraph list items.

This paragraph won't be part of the note, because it
isn't indented.
```

Footnotes

The above syntax generates the following output:

Here is a footnote reference,[1] and another.[2]

This paragraph won't be part of the note, because it isn't indented.

1. Knitr and ObservableJS also supported.

2. Here's one with multiple blocks.

   Subsequent paragraphs are indented to show that they belong to the previous footnote.

   ```
   { some.code }
   ```

   The whole paragraph can be indented, or just the first line. In this way, multi-paragraph footnotes work like multi-paragraph list items.

Footnotes

In addition, you can also write single paragraph footnotes inline using the following syntax:

```
Here is an inline note.^[Inlines notes are easier to write,
since you don't have to pick an identifier and move down to
type the note.]
```

This syntax generates the following output:

Here is an inline note.[1]

1. Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.

Footnotes

Footnote IDs should be unique

Footnote identifiers, e.g., the `1` in `^1`, need to be unique within a document. In Quarto books, chapters are combined into a single document for certain formats (including PDF, DOCX, and EPUB), so footnote identifiers need to be unique **across** chapters.

The footnotes that are generated from the above examples are included in the Example Footnotes section at the bottom of the page. See the Markdown Footnotes for additional information.

Tables

Markdown Syntax

```
| Right | Left | Default | Center |
|------:|:-----|---------|:------:|
|   12  |  12  |    12   |    12  |
|  123  |  123 |   123   |   123  |
|    1  |    1 |     1   |     1  |
```

Output

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

Learn more in the article on Tables.

Raw Content

Raw content can be included directly without Quarto parsing it using Pandoc's raw attribute. A raw block starts with ```{= followed by a format and closing }, e.g. here's a raw HTML block:

````
```{=html}
<iframe src="https://quarto.org/" width="500" height="400"></iframe>
```
````

You can also include raw content inline:

```
Here's some raw inline HTML: `<a>html</a>`{=html}
```

Equations

Use `$` delimiters for inline math and `$$` delimiters for display math.

For example:

Markdown Syntax	Output
`inline math: $E = mc^{2}$`	inline math: $E = mc^2$

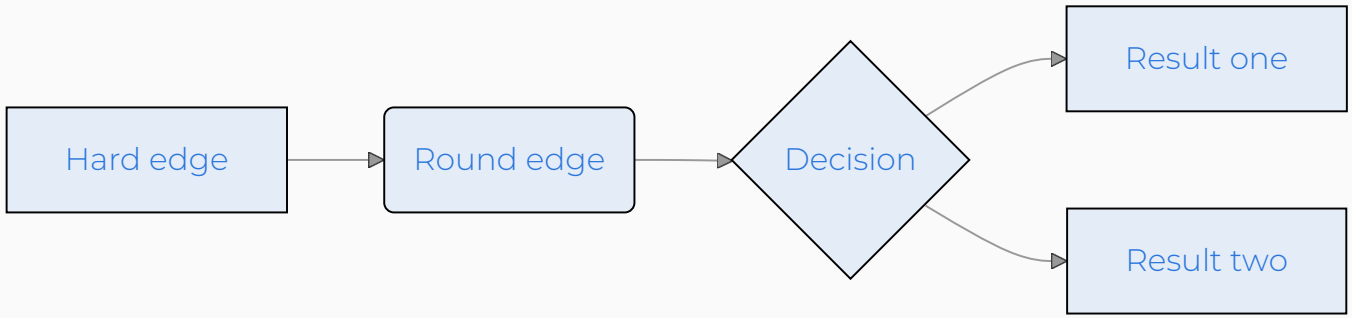
`display math:`
 	
`$$E = mc^{2}$$`
	display math:

Diagrams

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

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

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



Learn more in the article on Diagrams.

Videos

You can include videos in documents using the `{{< video >}}` shortcode. For example, here we embed a YouTube video:

```
{{< video https://www.youtube.com/embed/_f3latmOhew >}}
```

You can also use raw html code chunks:

````
```{=html}
<iframe width="560" height="315" src="https://www.youtube.com/embed/_f3latmOhew"></iframe>
```
````

Videos can refer to video files (e.g. MPEG) or can be links to videos published on YouTube, Vimeo, or Brightcove. Learn more in the article on Videos.

Divs and Spans

You can add classes, attributes, and other identifiers to regions of content using Divs and Spans (you'll see an example of this below in Callout Blocks).

For example, here we add the "border" class to a region of content using a div (`:::`):

```
::: {.border}
This content can be styled with a border
:::
```

Once rendered to HTML, Quarto will translate the markdown into:

```
<div class="border">
  <p>This content can be styled with a border</p>
</div>
```

Divs and Spans

Divs start with a fence containing at least three consecutive colons plus some attributes. The Div ends with another line containing a string of at least three consecutive colons. The Div should be separated by blank lines from preceding and following blocks.

Divs may also be nested. For example:

```
::::: {#special .sidebar}

::: {.warning}
Here is a warning.
:::

More content.
:::::
```

Once rendered to HTML, Quarto will translate the markdown into:

```
<div id="special" class="sidebar">
  <div class="warning">
    <p>Here is a warning.</p>
  </div>
  <p>More content.</p>
</div>
```

Divs and Spans

A bracketed sequence of inlines, as one would use to begin a link, will be treated as a `Span` with attributes if it is followed immediately by attributes:

```
[This is *some text*]{.class key="val"}
```

Once rendered to HTML, Quarto will translate the markdown into:

```
<span class="class" data-key="val">
  This is <em>some text</em>
</span>
```

Typically, you'll use CSS and/or a Filter along with Divs and Spans to provide styling or other behavior within rendered documents.

Ordering of Attributes

Both divs and spans in Pandoc can have any combination of identifiers, classes, and (potentially many) key-value attributes. In order for these to be recognized, they have to be provided in a specific order: identifiers, classes, and then key-value attributes. Any of these can be omitted, but must follow that order if they are provided. For example, the following is valid:

```
[This is good]{#id .class key1="val1" key2="val2"}
```

However, the following *will not be recognized*:

```
[This does *not* work!]{.class key="val" #id}
```

This ordering restriction applies to both divs and spans. See the documentation on Divs and Spans for additional details.

Callout Blocks

Markdown Syntax

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

Output

Note

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

Learn more in the article on Callout Blocks.

Other Blocks

```
> Blockquote
```

> Blockquote

```
::: {.classname}
Div
:::
```

Div

```
| Line Block
|   Spaces and newlines
|   are preserved
```

Line Block

Spaces and newlines

are preserved

Special Characters

Markdown Syntax	Output
`endash: --`	endash: –
`emdash: ---`	emdash: —

Keyboard Shortcuts

The `kbd` shortcode can be used to describe keyboard shortcuts in documentation. On Javascript formats, it will attempt to detect the operating system of the format and show the correct shortcut. On print formats, it will print the keyboard shortcut information for all operating systems.

For example, writing the following markdown:

```
1 To print, press {{< kbd Shift-Ctrl-P >}}. To open an existing new project, press {{< kbd mac=Sh
```

will render the keyboard shortcuts as:

To print, press **`Shift-Ctrl-P`**. To open an existing new project, press **`Shift-Command-O`**.

Markdown Example

Example of a markdown document…

```
## Introduction



Welcome to my **awesome** class. You
will learn all kinds of useful things
about Quarto.


- Markdown is simple
- You can add `python` code
```

Here's what the output looks like…

Introduction

Welcome to my **awesome** class. You will learn all kinds of useful things about Quarto.

- Markdown is simple
- You can add `python` code

Pretty Code

- Over 20 syntax highlighting themes available
- Default theme optimized for accessibility

pick_teams.py

```python
import random

def pick_teams(players: list) → dict:
    pass

names = ["Alice", "Bob", "Carol", "Dave"]
teams = pick_teams(names)
```

Code Animations

- Over 20 syntax highlighting themes available
- Default theme optimized for accessibility

pick_teams.py

```python
import random

def pick_teams(players: list) → dict:
    random.shuffle(players)  # Shuffle the list randomly once
    mid = len(players) // 2
    team_a = players[:mid]
    team_b = players[mid:]
    return {"Team A": team_a, "Team B": team_b}

names = ["Alice", "Bob", "Carol", "Dave"]
teams = pick_teams(names)
print(teams)
```

Line Highlighting

- Highlight specific lines for emphasis
- Incrementally highlight additional lines

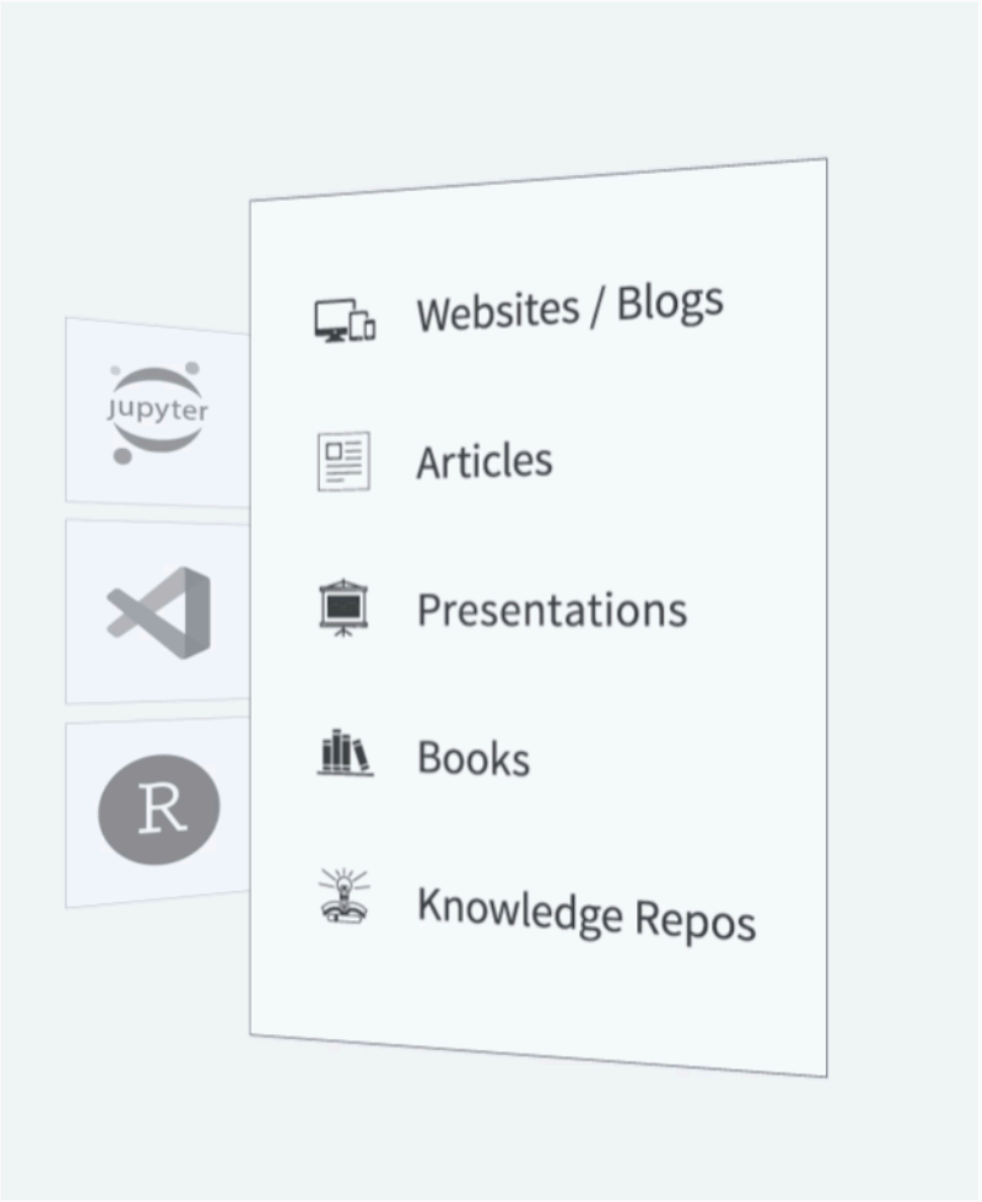
```
 1 import numpy as np
 2 import matplotlib.pyplot as plt
 3
 4 r = np.arange(0, 2, 0.01)
 5 theta = 2 * np.pi * r
 6 fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
 7 ax.plot(theta, r)
 8 ax.set_rticks([0.5, 1, 1.5, 2])
 9 ax.grid(True)
10 plt.show()
```

3 - Code

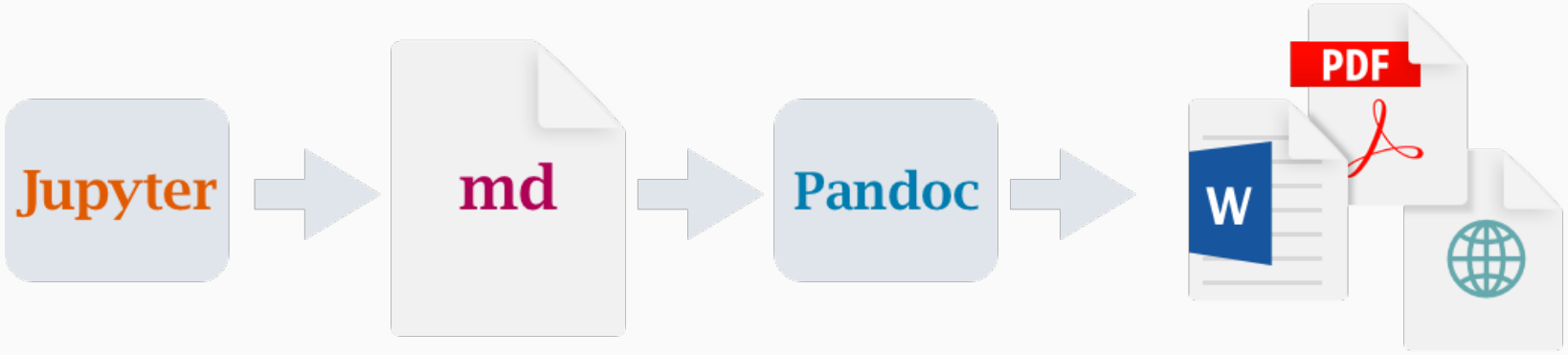
Quarto — https://quarto.org

An open-source scientific and technical publishing system that builds on standard markdown with features essential for scientific communication.

- Pandoc Markdown
- Jupyter Kernels
- Dozens of Output Formats
- Specialized Project Types
- Supported languages: Python, R, Julia, and Observable



How does Quarto work?



- Computations: Jupyter[1]
- Markdown: Pandoc w/ many enhancements
- Output: Documents, presentations, websites, books, blogs

1. Knitr and ObservableJS also supported.

Render and Preview

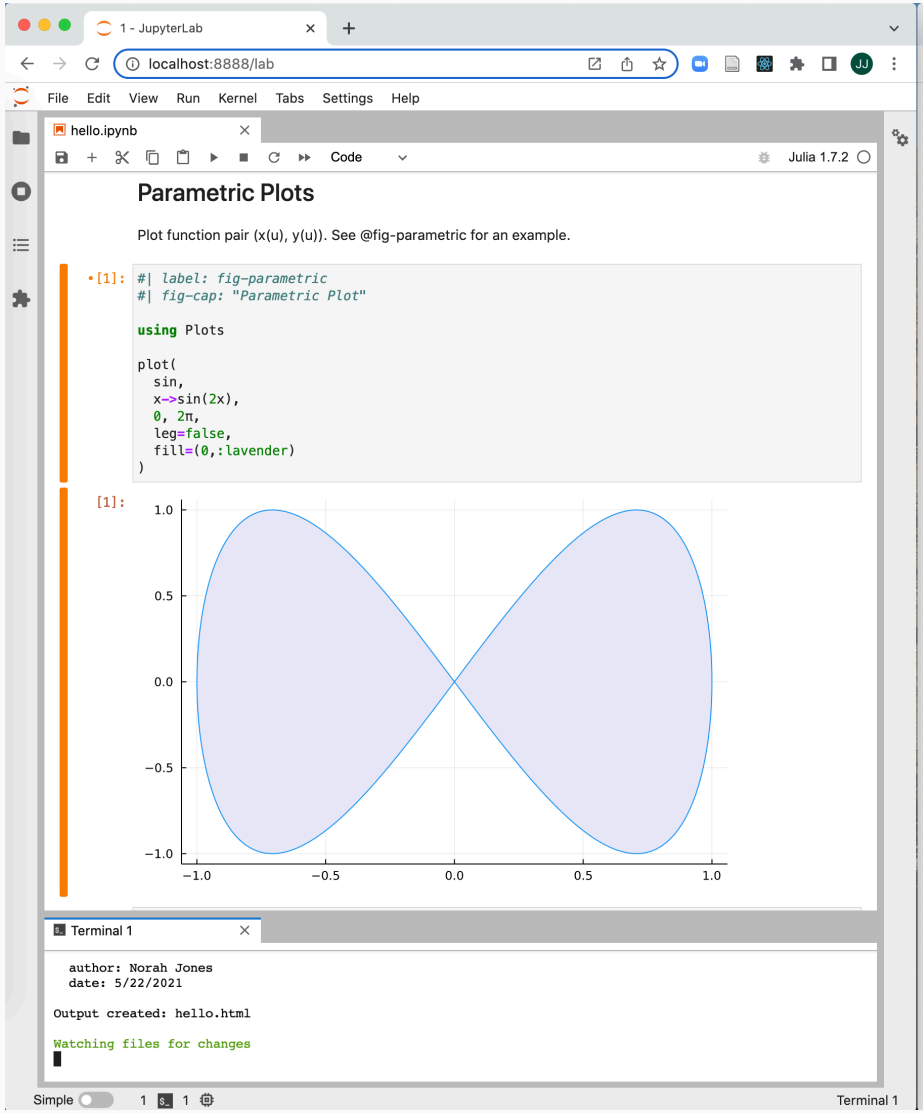
Render to output formats:

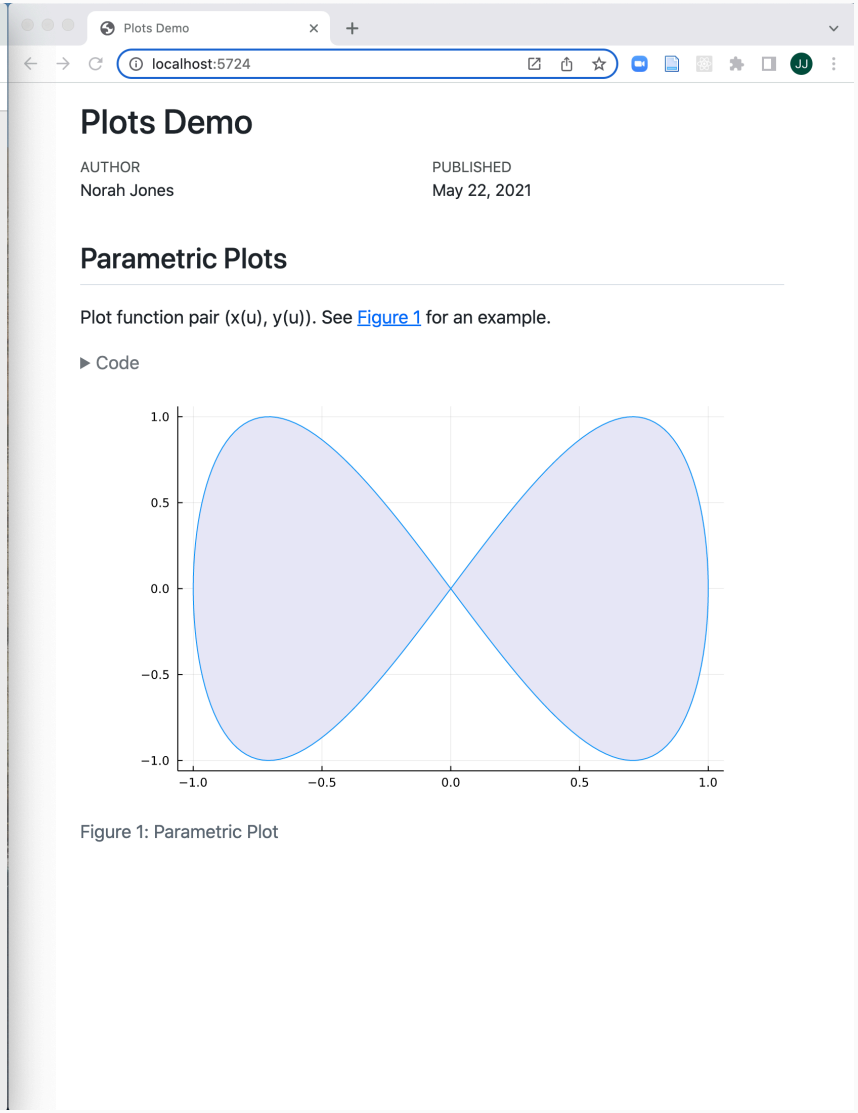
```
# ipynb notebook
quarto render notebook.ipynb
quarto render notebook.ipynb --to docx

# plain text qmd
quarto render notebook.qmd
quarto render notebook.qmd --to pdf
```

Live preview server (re-render on save):

```
# ipynb notebook
quarto preview notebook.ipynb
quarto preview notebook.ipynb --to docx

# plain text qmd
quarto preview notebook.qmd
quarto preview notebook.qmd --to pdf
```

Render and Preview





Plain Text Notebooks with .qmd Files

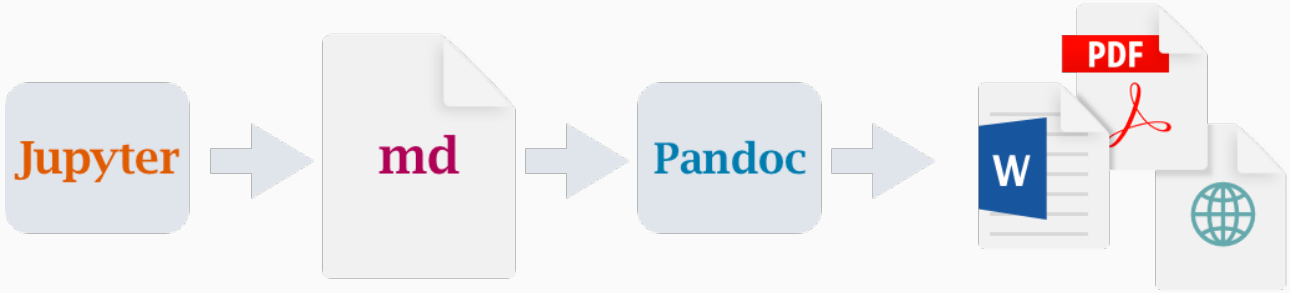
penguins.qmd

````
---
title: "Palmer Penguins"
author: Norah Jones
date: March 12, 2023
format: html
jupyter: python3
---

```{python}
#| echo: false

import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", \
         "bill_length_mm", "bill_depth_mm"]]
```

## Exploring the Data

See @fig-bill-sizes for an exploration of bill sizes.

```{python}
#| label: fig-bill-sizes
#| fig-cap: Bill Sizes by Species

import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm") \
  .add_legend()
```
````

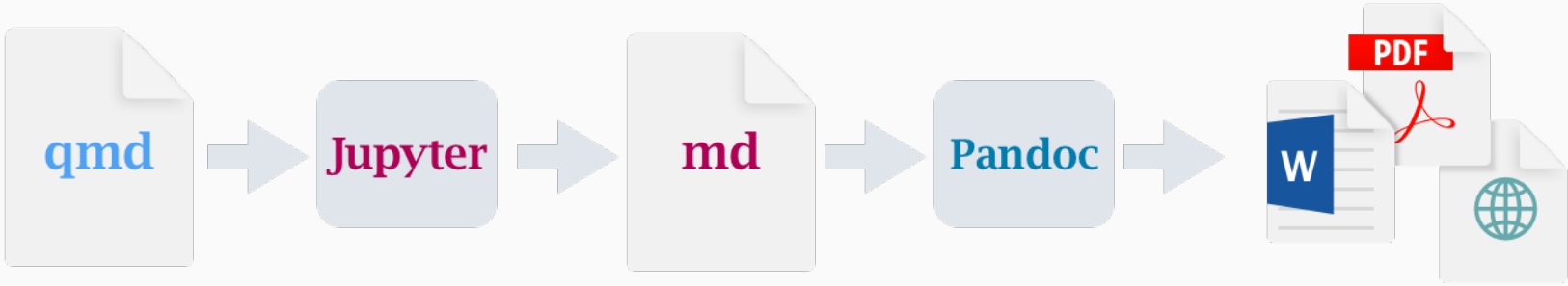
- Editable with any text editor (extensions for VS Code, Neovim, and Emacs)
- Cells always run in the same order
- Integrates well with version control
- Cache output with Jupyter Cache or Quarto freezer
- Lots of pros and cons visa-vi traditional `.ipynb` format/editors, use the right tool for each job

Rendering Pipeline

Notebook workflow (no execution occurs by default):



Plain text workflow (`.qmd` => `.ipynb` then execute cells):



Render Notebook to PDF

pdf.ipynb

Python 3 (ipykernel)

```
---
author: "Norah Jones"
date: March 22, 2023
format: pdf
bibliography: references.bib
citation-location: margin
---
```

[2]:
```
#| echo: false
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
```

Overview

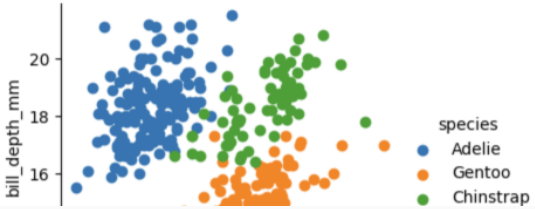
Here we will explore the use of the Palmer Penguins dataset from @palmerpenguins. The dataset contains different body
Antarctica.

Exploring

```
#| layout-ncol: 2
#| column: page
import matplotlib.pyplot as plt
import seaborn as sns
```

[7]:
```
#| layout-ncol: 2
#| column: page
import matplotlib.pyplot as plt
import seaborn as sns
sns.FacetGrid(df, hue="species", height=3, aspect=2.5/2) \
  .map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
sns.FacetGrid(df, hue="species", height=3, aspect=2.5/2) \
  .map(sns.kdeplot, "bill_length_mm",fill=True).add_legend()
```

[7]:
```
<seaborn.axisgrid.FacetGrid at 0x16cbd4460>
```



Palmer Penguins

Norah Jones

2023-03-22

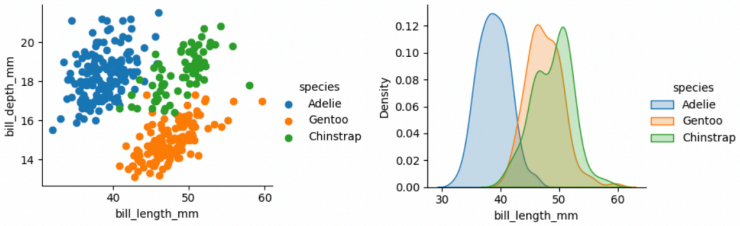
Overview

Here we will explore the use of the Palmer Penguins dataset from Horst, Hill, and Gorman (2020). The dataset contains different body measurements for three species of penguins from three islands in the Palmer Archipelago, Antarctica.

Horst, Allison Marie, Alison Presmanes Hill, and Kristen B Gorman. 2020. *Palmerpenguins: Palmer Archipelago (Antarctica) Penguin Data.* https://doi.org/10.5281/zenodo.3960218.

Exploring the Data

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.FacetGrid(df, hue="species", height=3, aspect=2.5/2) \
  .map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
sns.FacetGrid(df, hue="species", height=3, aspect=2.5/2) \
  .map(sns.kdeplot, "bill_length_mm",fill=True).add_legend()
```



Render Notebook to Revealjs

presentation.ipynb

Python 3 (ipykernel)

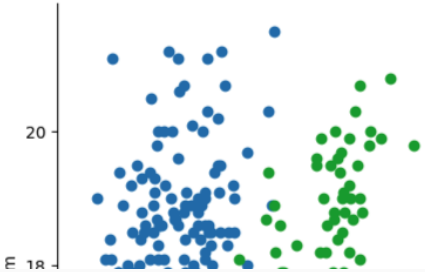
```
---
author: "Norah Jones"
date: "March 12, 2023"
format:
  revealjs:
    slide-number: c/t
    logo: palmer.png
---
```

[2]:
```
#| echo: false
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
```

Exploring the Data

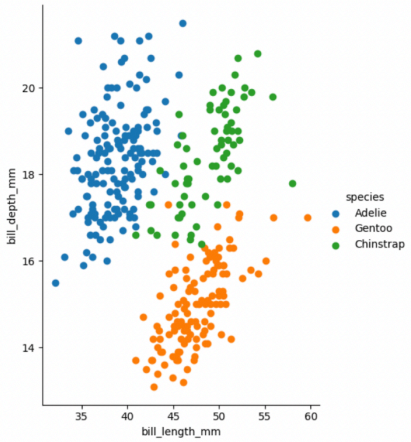
[7]:
```
#| layout-ncol: 2
import matplotlib.pyplot as plt
import seaborn as sns

sns.FacetGrid(df, hue="species", height=6, aspect=3/4) \
  .map(plt.scatter, "bill_length_mm", "bill_depth_mm") \
  .add_legend()

sns.FacetGrid(df, hue="species", height=6, aspect=3/4) \
  .map(sns.kdeplot, "bill_length_mm",fill=True) \
  .add_legend()
```

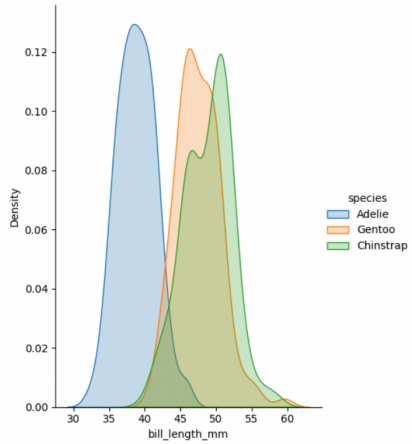
[7]:
```
<seaborn.axisgrid.FacetGrid at 0x137f56b60>
```



2 / 3

Exploring the Data





Render Notebook to HTML

default options

basics.ipynb

Markdown

Python 3 (ipykernel)

Palmer Penguins

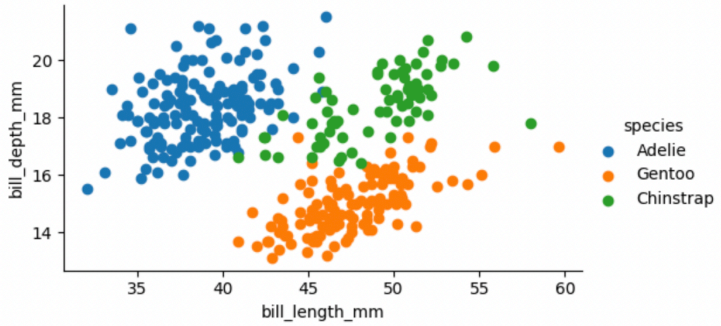
[9]:
```
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
df.head(3)
```

[9]:

	species	island	year	bill_length_mm	bill_depth_mm
0	Adelie	Torgersen	2007	39.1	18.7
1	Adelie	Torgersen	2007	39.5	17.4
2	Adelie	Torgersen	2007	40.3	18.0

[11]:
```
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```

[11]:
```
<seaborn.axisgrid.FacetGrid at 0x16c825300>
```

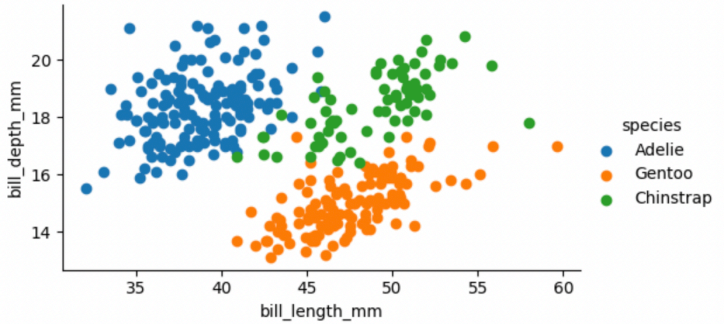


Palmer Penguins

```
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
df.head(3)
```

	species	island	year	bill_length_mm	bill_depth_mm
0	Adelie	Torgersen	2007	39.1	18.7
1	Adelie	Torgersen	2007	39.5	17.4
2	Adelie	Torgersen	2007	40.3	18.0

```
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```



Render Notebook to HTML

document level options

doc-options.ipynb

Python 3 (ipykernel)

```
---
theme: minty
highlight-style: atom-one
comments:
  hypothesis: true
---
```

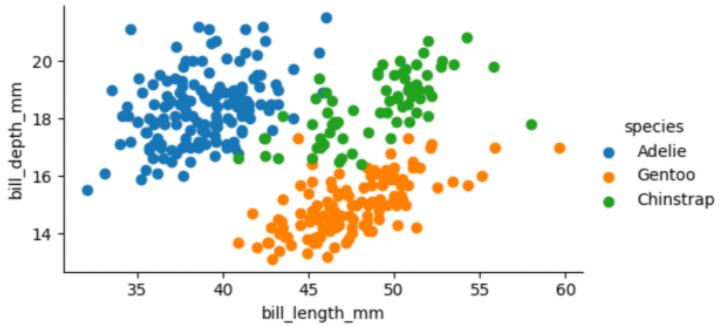
[9]:
```
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
df.head(3)
```

[9]:

	species	island	year	bill_length_mm	bill_depth_mm
0	Adelie	Torgersen	2007	39.1	18.7
1	Adelie	Torgersen	2007	39.5	17.4
2	Adelie	Torgersen	2007	40.3	18.0

[11]:
```
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```

[11]:
```
<seaborn.axisgrid.FacetGrid at 0x16c825300>
```

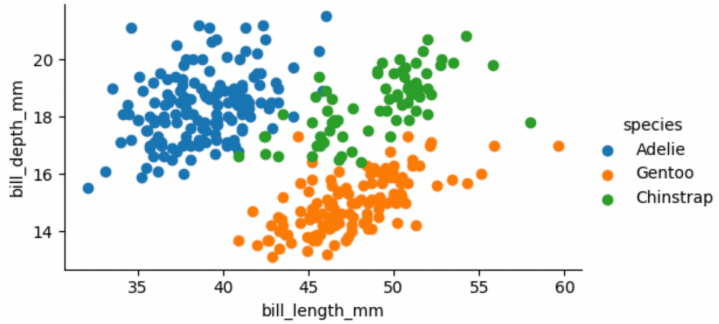


Palmer Penguins

```
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
df.head(3)
```

	species	island	year	bill_length_mm	bill_depth_mm
0	Adelie	Torgersen	2007	39.1	18.7
1	Adelie	Torgersen	2007	39.5	17.4
2	Adelie	Torgersen	2007	40.3	18.0

```
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```



Render Notebook to HTML

document and cell level options

cell-options.ipynb

Python 3 (ipykernel)

```
---
author: "Norah Jones"
date: "March 12, 2023"
code-tools: true
code-fold: true
---
```

Palmer

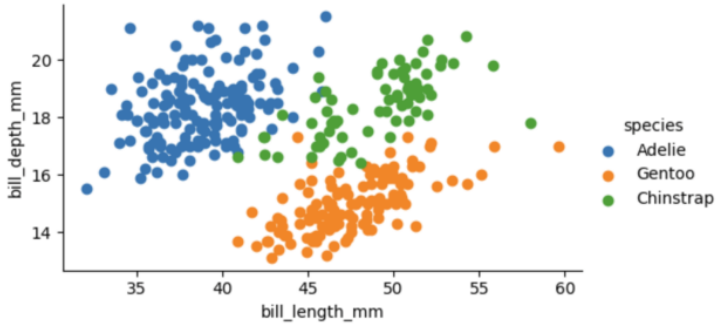
```
#| echo: false
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
```

Exploring

See @fig-bill-s

```
#| label: fig-bill-sizes
#| fig-cap: "Bill Sizes by Species"
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```

```
<seaborn.axisgrid.FacetGrid at 0x10612e080>
```



Palmer Penguins

</> Code

AUTHOR
Norah Jones

PUBLISHED
March 12, 2023

Exploring the Data

See Figure 1 for an exploration of bill sizes by species.

▶ Code

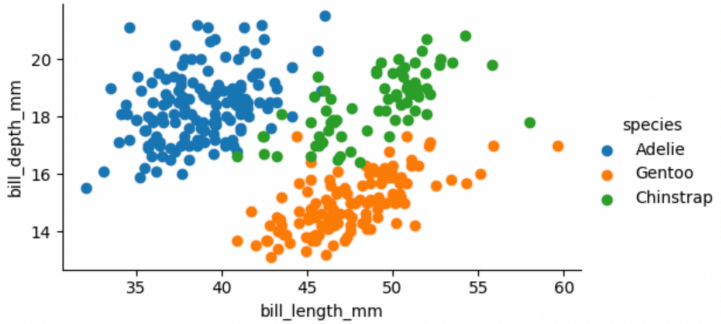


Figure 1: Bill Sizes by Species

Render Notebook to HTML

document and cell level options

cell-options.ipynb

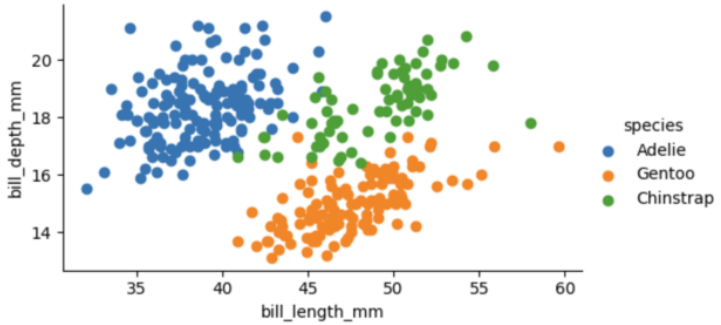
Python 3 (ipykernel)

```
---
author: "Norah Jones"
date: "March 12, 2023"
code-tools: true
code-fold: true
---
```

[1]:
```
#| echo: false
import pandas as pd
df = pd.read_csv("palmer-penguins.csv")
df = df[["species", "island", "year", "bill_length_mm", "bill_depth_mm"]]
```

```
#| label: fig-bill-sizes
#| fig-cap: "Bill Sizes by Species"
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```

[2]: `<seaborn.axisgrid.FacetGrid at 0x10612e080>`



Palmer Penguins

Code

AUTHOR
Norah Jones

PUBLISHED
March 12, 2023

Show All Code
Hide All Code

Exploring the Data

See Figure 1 for an exploration of bill sizes by species.

▼ Code

```
import matplotlib.pyplot as plt
import seaborn as sns
g = sns.FacetGrid(df, hue="species", height=3, aspect=3.5/2)
g.map(plt.scatter, "bill_length_mm", "bill_depth_mm").add_legend()
```

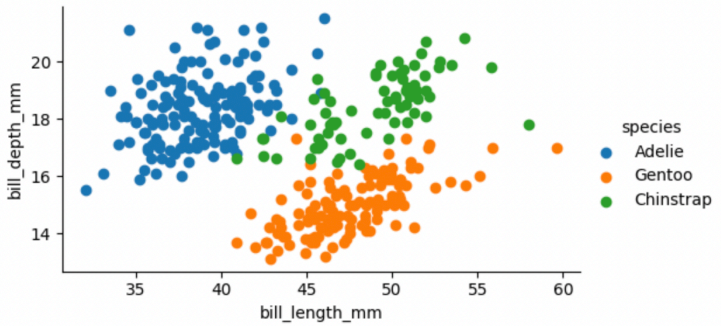


Figure 1: Bill Sizes by Species

javascript:void(0)

Notebook → Dashboard

gapminder-notebook.ipynb

Raw

Python 3 (ipykernel)

```
---
title: "Development Indicators by Continent"
author: "Gapminder Analytics Group"
format: dashboard
---
```

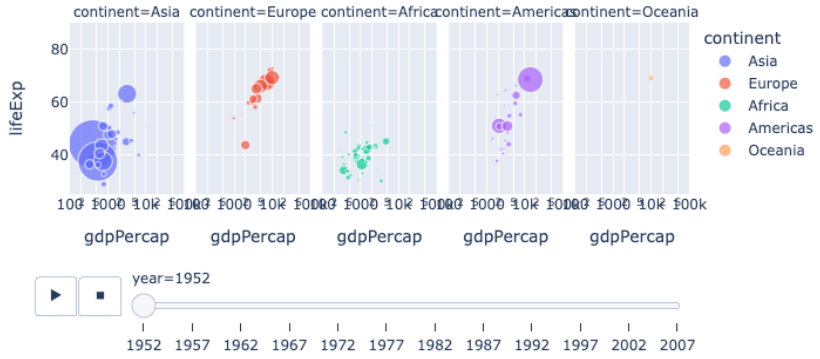
[1]:
```
import plotly.express as px
df = px.data.gapminder()
```

Indicators

Row

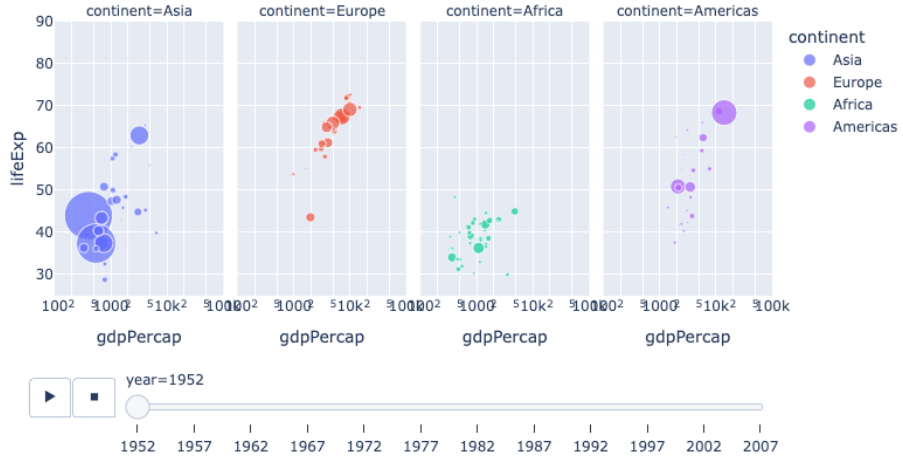
height="60%"

[2]:
```
#| title: GDP and Life Expectancy
px.scatter(
  df, x="gdpPercap", y="lifeExp",
  animation_frame="year", animation_group="country",
  size="pop", color="continent", hover_name="country",
  facet_col="continent", log_x=True, size_max=45,
  range_x=[100,100000], range_y=[25,90]
)
```

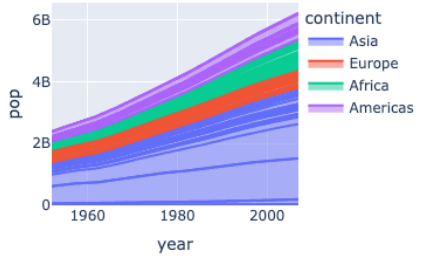


Worldwide Development Gapminder Analytics | Indicators | Data

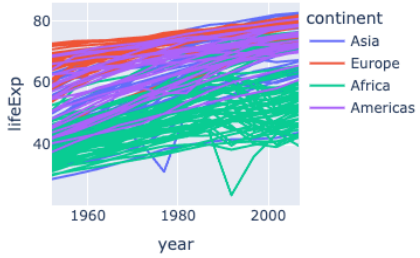
GDP and Life Expectancy



Population



Life Expectancy



Notebook → Dashboard

gapminder-notebook.ipynb

Raw

Python 3 (ipykernel)

Data

GAPMINDER Learn more about the Gapminder dataset at
https://www.gapminder.org/data/documentation/

[9]:
```
from itables import show
show(df, showIndex = False)
```

Show 10 entries Search:

country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	is
Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	
Afghanistan	Asia	1957	30.332	9240934	820.85303	AFG	
Afghanistan	Asia	1962	31.997	10267083	853.10071	AFG	
Afghanistan	Asia	1967	34.02	11537966	836.197138	AFG	
Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	
Afghanistan	Asia	1977	38.438	14880372	786.11336	AFG	
Afghanistan	Asia	1982	39.854	12881816	978.011439	AFG	
Afghanistan	Asia	1987	40.822	13867957	852.395945	AFG	
Afghanistan	Asia	1992	41.674	16317921	649.341395	AFG	
Afghanistan	Asia	1997	41.763	22227415	635.341351	AFG	

Showing 1 to 10 of 1,024 entries (downsampled from 1,704x8 to 1,024x8 as maxBytes=65536)

Previous 1 2 3 4 5 ... 103 Next

[ ]:

Worldwide Development Gapminder Analytics Indicators Data

GAPMINDER Learn more about the Gapminder dataset at
https://www.gapminder.org/data/documentation/

Showing 1,680 entries Search:

country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_nu
Afghanistan	Asia	1952	28.801	8425333	779.445314	AFG	4
Afghanistan	Asia	1957	30.332	9240934	820.85303	AFG	4
Afghanistan	Asia	1962	31.997	10267083	853.10071	AFG	4
Afghanistan	Asia	1967	34.02	11537966	836.197138	AFG	4
Afghanistan	Asia	1972	36.088	13079460	739.981106	AFG	4
Afghanistan	Asia	1977	38.438	14880372	786.11336	AFG	4
Afghanistan	Asia	1982	39.854	12881816	978.011439	AFG	4
Afghanistan	Asia	1987	40.822	13867957	852.395945	AFG	4
Afghanistan	Asia	1992	41.674	16317921	649.341395	AFG	4
Afghanistan	Asia	1997	41.763	22227415	635.341351	AFG	4
Afghanistan	Asia	2002	42.129	25268405	726.734055	AFG	4
Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
Albania	Europe	1952	55.23	1282697	1601.056136	ALB	8
Albania	Europe	1957	59.28	1476505	1942.284244	ALB	8
Albania	Europe	1962	64.82	1728137	2312.888958	ALB	8
Albania	Europe	1967	66.22	1984060	2760.196931	ALB	8
Albania	Europe	1972	67.69	2263554	3313.422188	ALB	8
Albania	Europe	1977	68.93	2509048	3533.00391	ALB	8
Albania	Europe	1982	70.42	2780097	3630.880722	ALB	8

Hands-on

Together: Checking / Installing Quarto

In VS Code or Positron, open a terminal and install Quarto from the command line:

Terminal

```
quarto check
```

Verify if you have Latex installed

Terminal

```
quarto list tools
```

Important

If you don't have Quarto installed, follow the Pre-Seminar Setup

Activity 1

Multi-Format Publishing: Researcher Profile

- **Objective**: Create a single Quarto document that describes yourself and your research, then render it to multiple formats.
- **Duration**: 15-20 minutes
- **Prerequisites**: Basic Quarto installation with latex (tinytex) and fundamental Markdown knowledge
  - https://cedgrueau.github.io/seminario-meim-2025.github.io/materials/hands-on/research-profile.html

20:00

Activity 2

Scientific Paper Writing with Quarto Extensions

- **Objective**: Learn how to use Quarto extensions to create professionally formatted scientific papers. Overview

> In this activity, you'll explore how Quarto extensions can transform your document into a properly formatted scientific paper. We'll use the IEEE template, one of the most common formats in computer science publications.

- https://cedgrueau.github.io/seminario-meim-2025.github.io/materials/hands-on/research-paper.html

15:00

Activity 1

Data visualization with Quarto and Observable JavaScript

- **Objective**: reate a Quarto document that downloads sample data from a URL and creates interactive visualizations using pure Observable JavaScript.
  - https://cedgrueau.github.io/seminario-meim-2025.github.io/materials/hands-on/data-visualization.html

10:00