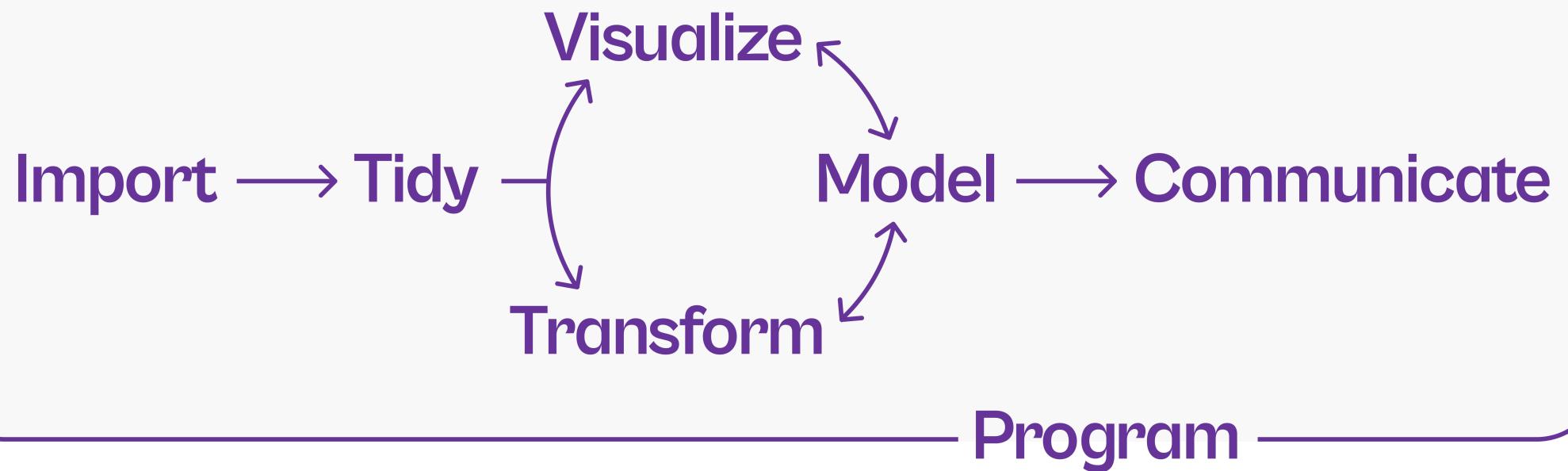


Reproducible Data Analysis with R

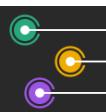
— Communicate Results with Quarto —

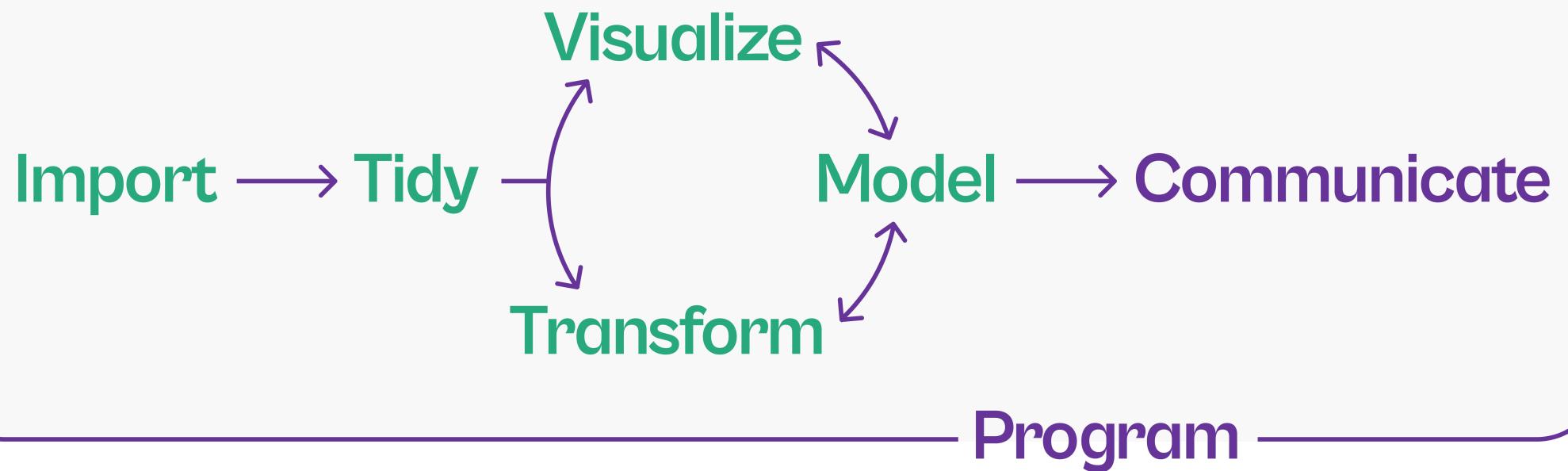
Cédric Scherer // R Course TU Dresden // Feb 27-Mar 3, 2023



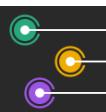


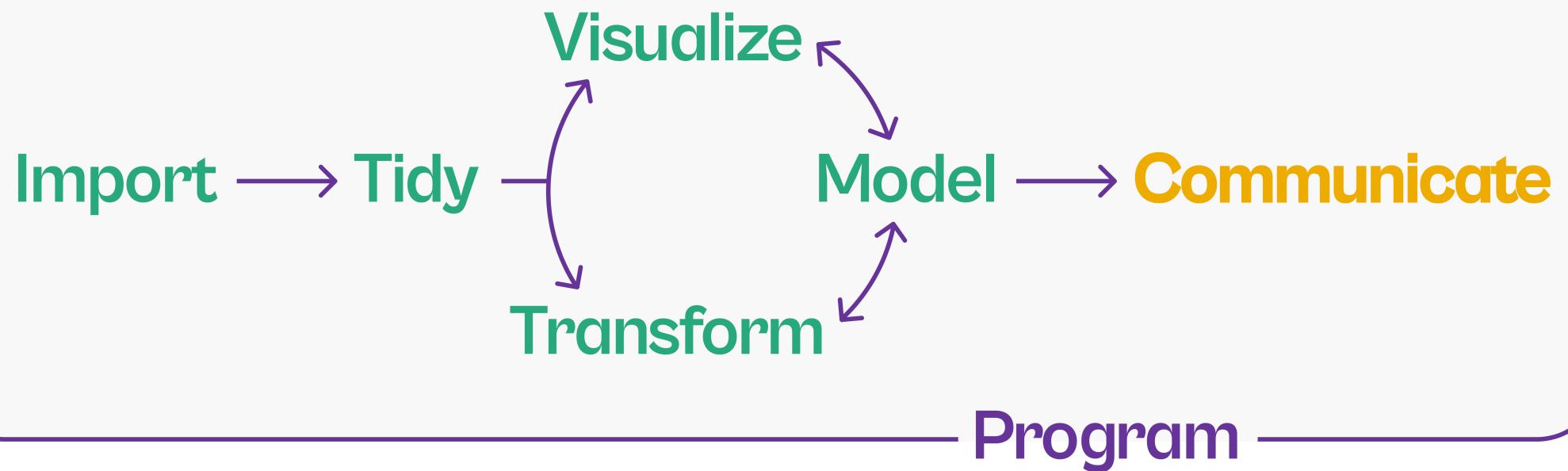
The data science workflow, modified from "[R for Data Science](#)"



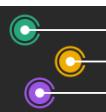


The data science workflow, modified from "R for Data Science"

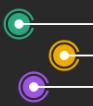


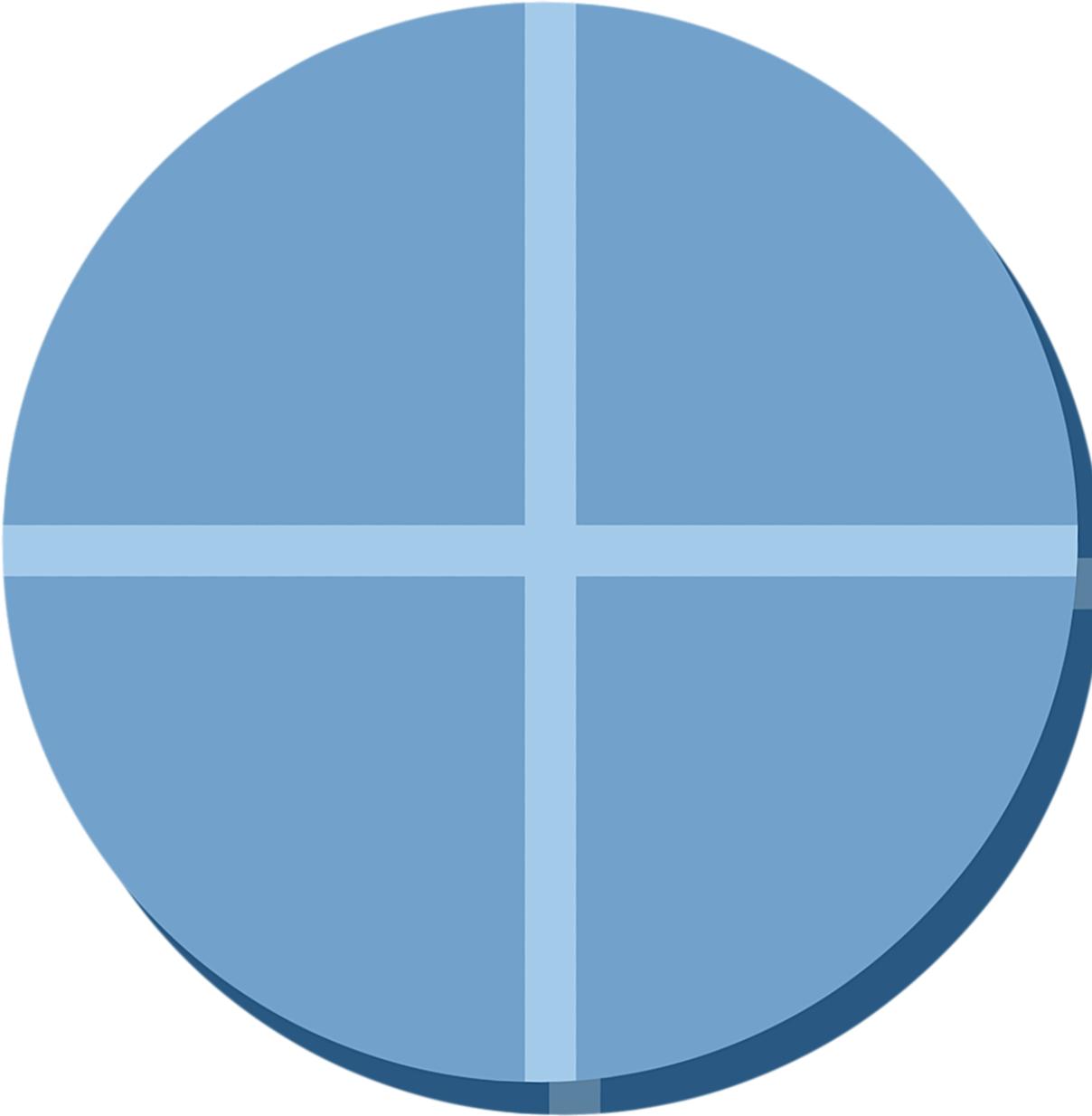


The data science workflow, modified from “R for Data Science”



Communicate Results with Quarto





Quarto® is an open-source scientific and technical publishing system.

You can **combine narrative text, code—and code outputs**.

You can **produce reports, presentations, web pages, books and more**.

You can **include tables*, visualizations*, animations and more**.

* even interactive ones!

You can **automate and scale the report generation process**.



Go to file/function Addins

hello.qmd x Render on Save ABC Render Run Edit

Source Visual Normal 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.

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

```
{r}
#| label: plot-penguins
```

(Top Level) Quarto

Console

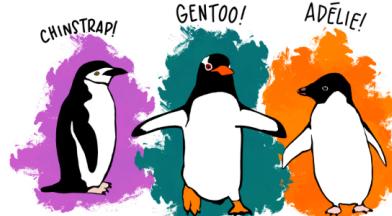
Hello, Quarto

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>.

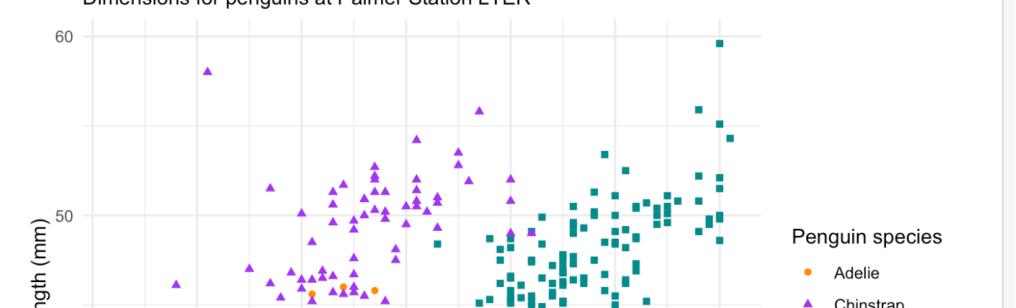
Meet the penguins

The `penguins` data from the [palmerpenguins](#) package contains size measurements for 344 penguins from three species observed on three islands in the Palmer Archipelago, Antarctica.



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

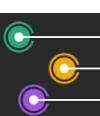
Flipper and bill length
Dimensions for penguins at Palmer Station LTER



Penguin species

- Adelie
- Chinstrap

Source: [Quarto Documentation](#) by Posit



authoring.qmd x Go to file/function Addins

authoring.qmd x Render on Save ABC Render Run Outline

Source Visual B I Normal Format Insert Table Outline

```
---  
title: "Housing Prices"  
author: "Mine Çetinkaya-Rundel"  
format: pdf  
---
```

Introduction
Exploratory data ...
Data visualization
Summary statis...
Modeling

Introduction

In this analysis, we build a model predicting sale prices of houses based on data on houses that were sold in the Duke Forest neighborhood of Durham, NC around November 2020. Let's start by loading the packages we'll use for the analysis.

```
{r}  
#| label: load-pkgs  
#| code-summary: "Packages"  
#| message: false  
  
library(openintro) # for data  
library(tidyverse) # for data wrangling and visualization  
library(knitr) # for tables  
library(broom) # for model summary
```

We present the results of exploratory data analysis in `@sec-eda` and the regression model in `@sec-model`.

```
<!--# ADD CITATION HERE -->
```

Exploratory data analysis

The data contains `r nrow(duke_forest)` houses. As part of the exploratory analysis let's visualize and summarize the relationship

Chunk 1 Quarto

Console

Environment History Connections Build Git Tutorial

Files Plots Packages Help Viewer Presentation

Publish

1 of 4 Automatic Zoom

Housing Prices

Mine Çetinkaya-Rundel

Introduction

In this analysis, we build a model predicting sale prices of houses based on data on houses that were sold in the Duke Forest neighborhood of Durham, NC around November 2020. Let's start by loading the packages we'll use for the analysis.

```
library(openintro) # for data  
library(tidyverse) # for data wrangling and visualization  
library(knitr) # for tables  
library(broom) # for model summary
```

We present the results of exploratory data analysis in Section and the regression model in Section .

Exploratory data analysis

The data contains 98 houses. As part of the exploratory analysis let's visualize and summarize the relationship between areas and prices of these houses.

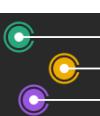
Data visualization

Figure 1 shows two histograms displaying the distributions of `price` and `area` individually.

```
ggplot(duke_forest, aes(x = price)) +  
  geom_histogram(binwidth = 50000) +  
  labs(title = "Histogram of prices")  
  
ggplot(duke_forest, aes(x = area)) +  
  geom_histogram(binwidth = 250) +  
  labs(title = "Histogram of areas")
```

Source: Quarto Documentation by Posit

Cédric Scherer // R Course TU Dresden // Communicating with Quarto



The screenshot shows the RStudio interface with a Quarto project open. The left pane displays the Quarto source code for 'index.qmd' and '_quarto.yml'. The right pane shows the rendered 'mywebsite' website.

Quarto Source (Left):

```
title: "mywebsite"
```

This is a Quarto website.
To learn more about Quarto websites visit
<https://quarto.org/docs/websites>.

Quarto Rendered Website (Right):

mywebsite

This is a Quarto website.
To learn more about Quarto websites visit
<https://quarto.org/docs/websites>.

(Top Level) Quarto

Console

Source: [Quarto Documentation](#) by Posit

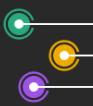
Cédric Scherer // R Course TU Dresden // Communicating with Quarto



What About RMarkdown?

RMarkdown is a documenting system by Posit and the **predecessor of Quarto**.

- same idea of creating reproducible and data-driven notebooks and reports
- combines the popular and simple Markdown syntax with R code
- different output formats and purposes



RMarkdown vs Quarto

RMarkdown:

- language-specific library
- designed for R users
- tied to R and RStudio IDE
- multiple packages for different outputs

Quarto:

- external software application
- multi-language support (R, Python, JS, Julia)
- multi-engine support (Knitr, Jupyter, Observable)
- single consistent ecosystem for all purposes

R Markdown is used extensively and will continue to be actively supported!

Quarto is the tool-agnostic “next-generation” RMarkdown.



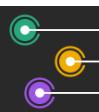
Quarto: Next-Generation RMarkdown

RMarkdown grew into a large ecosystem, with **varying syntax**.

Quarto comes “**batteries included**” straight out of the box

- HTML reports and websites
- PDF reports
- MS Office (Word, Powerpoint)
- Presentations (Powerpoint, Beamer, `revealjs`)
- Books

Any language, *exact* same approach and syntax



Quarto: Next-Generation RMarkdown

Quarto is a command line interface (CLI)—not an R package—that renders plain text formats* or mixed formats** into reports, books, presentations, websites,

* such as `.qmd`, `.rmd`, `.md`

** such as `.ipynb`



Quarto: Next-Generation RMarkdown

Quarto is a command line interface (CLI)—not an R package—that renders plain text formats* or mixed formats** into reports, books, presentations, websites,

* such as `.qmd`, `.rmd`, `.md`

** such as `.ipynb`

Note that there is an `{quarto}` R package, too. However, it “only” provides convenient functions for command line rendering from R, and is not required for using the Quarto CLI with R.



Comfort of Your Own Workspace

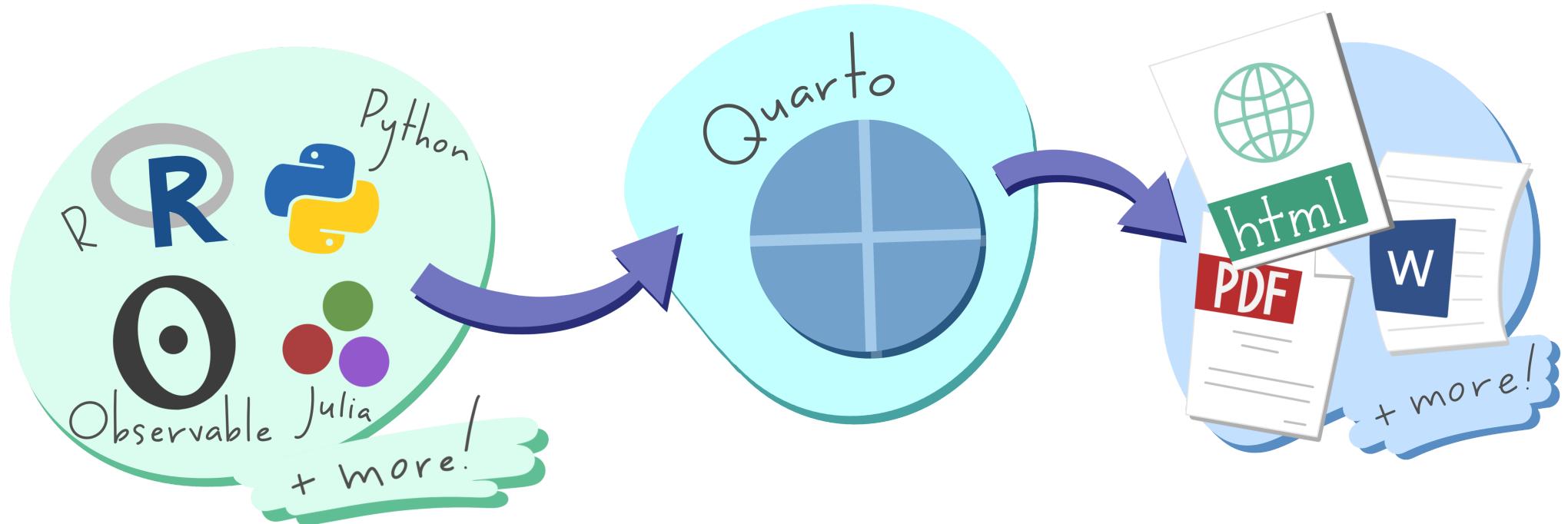


Illustration by Allison Horst



Comfort of Your Own Workspace

The screenshot shows the Quarto RStudio interface. On the left, the code editor displays a Quarto document with R code for generating a scatter plot and summarizing the airquality dataset. The right side shows the generated output: a scatter plot of Ozone vs Temp with a loess smoothing line, and a table of summary statistics for the airquality dataset.

```
quarto-rstudio.qmd
```

```
1 ---  
2 title: "Knitr Document"  
3 format:  
4   html:  
5     code-background: true  
6 execute:  
7   warning: false  
8 ---  
9  
10 ````{r}  
11 library(ggplot2)  
12 ggplot(airquality, aes(Temp, Ozone)) +  
13   geom_point() +  
14   geom_smooth(method = "loess", se = FALSE)  
15 ````  
16  
17 ````{r}  
18 summary(airquality)  
19 ````
```

Environment History Connections Build Tutorial

Knitr Document

```
library(ggplot2)  
ggplot(airquality, aes(Temp, Ozone)) +  
  geom_point() +  
  geom_smooth(method = "loess", se = FALSE)
```

Ozone

Temp

Ozone	Solar.R	Wind	Temp
40	8.0	7.4	69.0
42	10.0	7.8	70.0
36	12.0	8.0	71.0
36	12.0	8.0	72.0
48	12.0	8.0	73.0
36	12.0	8.0	74.0
36	12.0	8.0	75.0
42	12.0	8.0	76.0
36	12.0	8.0	77.0
42	12.0	8.0	78.0
36	12.0	8.0	79.0
42	12.0	8.0	80.0
36	12.0	8.0	81.0
42	12.0	8.0	82.0
36	12.0	8.0	83.0
42	12.0	8.0	84.0
36	12.0	8.0	85.0
42	12.0	8.0	86.0
36	12.0	8.0	87.0
42	12.0	8.0	88.0
36	12.0	8.0	89.0
42	12.0	8.0	90.0
36	12.0	8.0	91.0
42	12.0	8.0	92.0
36	12.0	8.0	93.0
42	12.0	8.0	94.0
36	12.0	8.0	95.0
42	12.0	8.0	96.0
36	12.0	8.0	97.0
42	12.0	8.0	98.0
36	12.0	8.0	99.0
42	12.0	8.0	100.0
36	12.0	8.0	101.0
42	12.0	8.0	102.0
36	12.0	8.0	103.0
42	12.0	8.0	104.0
36	12.0	8.0	105.0
42	12.0	8.0	106.0
36	12.0	8.0	107.0
42	12.0	8.0	108.0
36	12.0	8.0	109.0
42	12.0	8.0	110.0
36	12.0	8.0	111.0
42	12.0	8.0	112.0
36	12.0	8.0	113.0
42	12.0	8.0	114.0
36	12.0	8.0	115.0
42	12.0	8.0	116.0
36	12.0	8.0	117.0
42	12.0	8.0	118.0
36	12.0	8.0	119.0
42	12.0	8.0	120.0
36	12.0	8.0	121.0
42	12.0	8.0	122.0
36	12.0	8.0	123.0
42	12.0	8.0	124.0
36	12.0	8.0	125.0
42	12.0	8.0	126.0
36	12.0	8.0	127.0
42	12.0	8.0	128.0
36	12.0	8.0	129.0
42	12.0	8.0	130.0
36	12.0	8.0	131.0
42	12.0	8.0	132.0
36	12.0	8.0	133.0
42	12.0	8.0	134.0
36	12.0	8.0	135.0
42	12.0	8.0	136.0
36	12.0	8.0	137.0
42	12.0	8.0	138.0
36	12.0	8.0	139.0
42	12.0	8.0	140.0
36	12.0	8.0	141.0
42	12.0	8.0	142.0
36	12.0	8.0	143.0
42	12.0	8.0	144.0
36	12.0	8.0	145.0
42	12.0	8.0	146.0
36	12.0	8.0	147.0
42	12.0	8.0	148.0
36	12.0	8.0	149.0
42	12.0	8.0	150.0
36	12.0	8.0	151.0
42	12.0	8.0	152.0
36	12.0	8.0	153.0
42	12.0	8.0	154.0
36	12.0	8.0	155.0
42	12.0	8.0	156.0
36	12.0	8.0	157.0
42	12.0	8.0	158.0
36	12.0	8.0	159.0
42	12.0	8.0	160.0
36	12.0	8.0	161.0
42	12.0	8.0	162.0
36	12.0	8.0	163.0
42	12.0	8.0	164.0
36	12.0	8.0	165.0
42	12.0	8.0	166.0
36	12.0	8.0	167.0
42	12.0	8.0	168.0
36	12.0	8.0	169.0
42	12.0	8.0	170.0
36	12.0	8.0	171.0
42	12.0	8.0	172.0
36	12.0	8.0	173.0
42	12.0	8.0	174.0
36	12.0	8.0	175.0
42	12.0	8.0	176.0
36	12.0	8.0	177.0
42	12.0	8.0	178.0
36	12.0	8.0	179.0
42	12.0	8.0	180.0
36	12.0	8.0	181.0
42	12.0	8.0	182.0
36	12.0	8.0	183.0
42	12.0	8.0	184.0
36	12.0	8.0	185.0
42	12.0	8.0	186.0
36	12.0	8.0	187.0
42	12.0	8.0	188.0
36	12.0	8.0	189.0
42	12.0	8.0	190.0
36	12.0	8.0	191.0
42	12.0	8.0	192.0
36	12.0	8.0	193.0
42	12.0	8.0	194.0
36	12.0	8.0	195.0
42	12.0	8.0	196.0
36	12.0	8.0	197.0
42	12.0	8.0	198.0
36	12.0	8.0	199.0
42	12.0	8.0	200.0
36	12.0	8.0	201.0
42	12.0	8.0	202.0
36	12.0	8.0	203.0
42	12.0	8.0	204.0
36	12.0	8.0	205.0
42	12.0	8.0	206.0
36	12.0	8.0	207.0
42	12.0	8.0	208.0
36	12.0	8.0	209.0
42	12.0	8.0	210.0
36	12.0	8.0	211.0
42	12.0	8.0	212.0
36	12.0	8.0	213.0
42	12.0	8.0	214.0
36	12.0	8.0	215.0
42	12.0	8.0	216.0
36	12.0	8.0	217.0
42	12.0	8.0	218.0
36	12.0	8.0	219.0
42	12.0	8.0	220.0
36	12.0	8.0	221.0
42	12.0	8.0	222.0
36	12.0	8.0	223.0
42	12.0	8.0	224.0
36	12.0	8.0	225.0
42	12.0	8.0	226.0
36	12.0	8.0	227.0
42	12.0	8.0	228.0
36	12.0	8.0	229.0
42	12.0	8.0	230.0
36	12.0	8.0	231.0
42	12.0	8.0	232.0
36	12.0	8.0	233.0
42	12.0	8.0	234.0
36	12.0	8.0	235.0
42	12.0	8.0	236.0
36	12.0	8.0	237.0
42	12.0	8.0	238.0
36	12.0	8.0	239.0
42	12.0	8.0	240.0
36	12.0	8.0	241.0
42	12.0	8.0	242.0
36	12.0	8.0	243.0
42	12.0	8.0	244.0
36	12.0	8.0	245.0
42	12.0	8.0	246.0
36	12.0	8.0	247.0
42	12.0	8.0	248.0
36	12.0	8.0	249.0
42	12.0	8.0	250.0
36	12.0	8.0	251.0
42	12.0	8.0	252.0
36	12.0	8.0	253.0
42	12.0	8.0	254.0
36	12.0	8.0	255.0
42	12.0	8.0	256.0
36	12.0	8.0	257.0
42	12.0	8.0	258.0
36	12.0	8.0	259.0
42	12.0	8.0	260.0
36	12.0	8.0	261.0
42	12.0	8.0	262.0
36	12.0	8.0	263.0
42	12.0	8.0	264.0
36	12.0	8.0	265.0
42	12.0	8.0	266.0
36	12.0	8.0	267.0
42	12.0	8.0	268.0
36	12.0	8.0	269.0
42	12.0	8.0	270.0
36	12.0	8.0	271.0
42	12.0	8.0	272.0
36	12.0	8.0	273.0
42	12.0	8.0	274.0
36	12.0	8.0	275.0
42	12.0	8.0	276.0
36	12.0	8.0	277.0
42	12.0	8.0	278.0
36	12.0	8.0	279.0
42	12.0	8.0	280.0
36	12.0	8.0	281.0
42	12.0	8.0	282.0
36	12.0	8.0	283.0
42	12.0	8.0	284.0
36	12.0	8.0	285.0
42	12.0	8.0	286.0
36	12.0	8.0	287.0
42	12.0	8.0	288.0
36	12.0	8.0	289.0
42	12.0	8.0	290.0
36	12.0	8.0	291.0
42	12.0	8.0	292.0
36	12.0	8.0	293.0
42	12.0	8.0	294.0
36	12.0	8.0	295.0
42	12.0	8.0	296.0
36	12.0	8.0	297.0
42	12.0	8.0	298.0
36	12.0	8.0	299.0
42	12.0	8.0	300.0
36	12.0	8.0	301.0
42	12.0	8.0	302.0
36	12.0	8.0	303.0
42	12.0	8.0	304.0
36	12.0	8.0	305.0
42	12.0	8.0	306.0
36	12.0	8.0	307.0
42	12.0	8.0	308.0
36	12.0	8.0	309.0
42	12.0	8.0	310.0
36	12.0	8.0	311.0
42	12.0	8.0	312.0
36	12.0	8.0	313.0
42	12.0	8.0	314.0
36	12.0	8.0	315.0
42	12.0	8.0	316.0
36	12.0	8.0	317.0
42	12.0	8.0	318.0
36	12.0	8.0	319.0
42	12.0	8.0	320.0
36	12.0	8.0	321.0
42	12.0	8.0	322.0
36	12.0	8.0	323.0
42	12.0	8.0	324.0
36	12.0	8.0	325.0
42	12.0	8.0	326.0
36	12.0	8.0	327.0
42	12.0	8.0	328.0
36	12.0	8.0	329.0
42	12.0	8.0	330.0
36	12.0	8.0	331.0
42	12.0	8.0	332.0
36	12.0	8.0	333.0
42	12.0	8.0	334.0
36	12.0	8.0	335.0
42	12.0	8.0	336.0
36	12.0	8.0	337.0
42	12.0	8.0	338.0
36	12.0	8.0	339.0
42	12.0	8.0	340.0
36	12.0	8.0	341.0
42	12.0	8.0	342.0
36	12.0	8.0	343.0
42	12.0	8.0	344.0
36	12.0	8.0	345.0
42	12.0	8.0	346.0
36	12.0	8.0	347.0
42	12.0	8.0	348.0
36	12.0	8.0	349.0
42	12.0	8.0	350.0
36	12.0	8.0	351.0
42	12.0	8.0	352.0
36	12.		

Comfort of Your Own Workspace

The image shows two browser windows side-by-side. The left window is titled 'quarto-jupyterlab - JupyterLab' and displays a Quarto document named 'quarto-jupyterlab.ipynb'. The document content includes a YAML front matter section and a section titled 'Polar Axis' with a note about a line plot. Below this is a code cell [2] containing Python code to generate a polar plot, followed by the resulting polar plot image. The right window is titled 'Matplotlib Demo' and shows a Matplotlib line plot on a polar axis. The plot features concentric circles and a spiral line. The Matplotlib demo page also includes author information (Norah Smith) and a date (May 22nd, 2021).

File Edit View Run Kernel Tabs Settings Help

quarto-jupyterlab.ipynb x

localhost:8888/lab/tree/quarto-jupyterlab.ipynb

Polar Axis

For a demonstration of a line plot on a polar axis, see @fig-polar.

```
[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()
```

90° 135° 45°

author: Norah Smith
date: 'May 22nd, 2021'

Output created: quarto-jupyterlab.html

Watching files for changes

Simple 1 s. 1 Python 3 (ipykernel) | Idle Mode: Edit Ln 3, Col 1 quarto-jupyterlab.ipynb

Matplotlib Demo

Norah Smith
May 22nd, 2021

Polar Axis

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

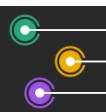
▶ Code

A polar plot with radial axes labeled at 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. Concentric circles are drawn at radii of 0.5, 1.0, 1.5, and 2.0. A blue spiral line starts at the origin and winds outwards, increasing in radius as it turns clockwise.

Figure 1: A line plot on a polar axis

Source: [Quarto Documentation](#) by Posit

Cédric Scherer // R Course TU Dresden // Communicating with Quarto



Comfort of Your Own Workspace

The screenshot shows a Jupyter Notebook interface with three main panes:

- EXPLORER** pane on the left, showing a file tree with several Quarto files (qmd) and a CSV file.
- Code Editor** pane in the center, displaying Python code for a "matplotlib demo". The code imports numpy and matplotlib.pyplot, creates a polar plot with concentric circles, and sets theta ticks at 45° intervals from 0° to 315°.
- Interactive Console** pane on the right, showing the Python environment and a generated polar plot.

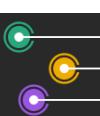
```
python.qmd
Users > jjallaire > Desktop > python.qmd > ...
1 ---  
2 title: "matplotlib demo"  
3 format:  
4   html:  
5     code-fold: true  
6 jupyter: python3  
7 ---  
8  
9 For a demonstration of a line plot on a  
polar axis, see @fig-polar.  
10  
11 > Run Cell  
12 ````{python}  
13 #| label: fig-polar  
14 #| fig-cap: "A line plot on a polar axis"  
15  
16 import numpy as np  
17 import matplotlib.pyplot as plt  
18  
19 r = np.arange(0, 2, 0.01)  
20 theta = 2 * np.pi * r  
21 fig, ax = plt.subplots(  
22   subplot_kw = {'projection': 'polar'}  
23 )  
24 ax.plot(theta, r)  
25 ax.set_rticks([0.5, 1, 1.5, 2])  
26 ax.grid(True)  
27 plt.show()  
````
```

The Interactive Console pane shows the Python environment and a generated polar plot:

Python 3.9.5 (v3.9.5:0a7dcdbb13, May 3 2021, 13:17:02)  
Type 'copyright', 'credits' or 'license' for more information  
IPython 7.25.0 -- An enhanced Interactive Python. Type '?' for help.

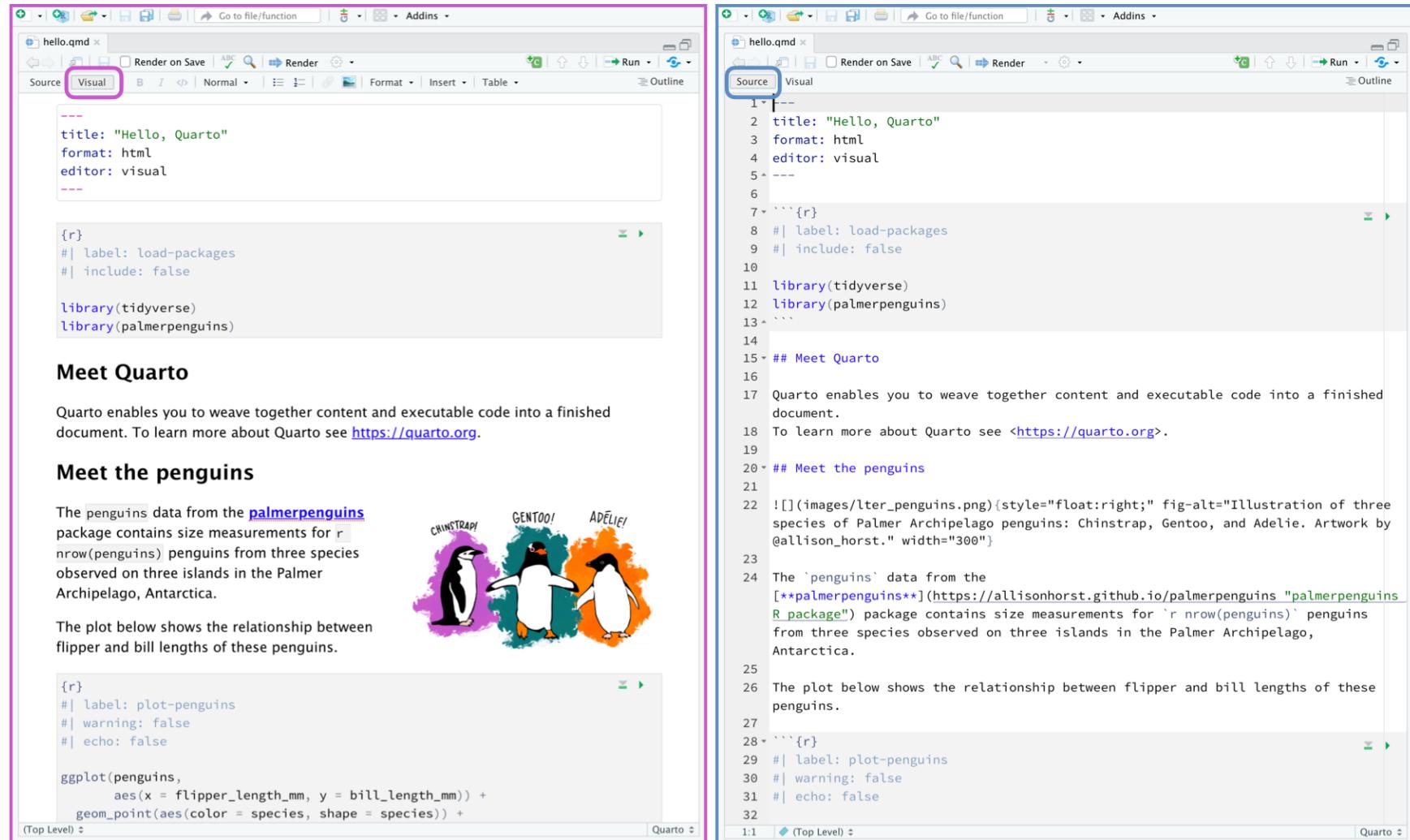
✓ import numpy as np ...

Source: [Quarto Documentation](#) by Posit



# Comfort of Your Own Workspace

In RStudio, you can write qmd/rmd reports using the visual or source editor:



The image shows two RStudio windows side-by-side, each displaying a Quarto document in different editors.

**Left Window (Visual Editor):**

- Editor Tab:** Visual (highlighted with a red box).
- Content:** Displays the front matter and code for a Quarto document named "hello.qmd". The code includes a title, format, and editor settings, followed by an R block that loads tidyverse and palmerpenguins packages.
- Content Below:** Includes sections titled "Meet Quarto" and "Meet the penguins", along with descriptive text and a small illustration of three penguins labeled CHINSTRAP, GENTOO!, and ADÉLIE!.
- Code Block:** Shows an R block for plotting penguins.

**Right Window (Source Editor):**

- Editor Tab:** Source (highlighted with a red box).
- Content:** Displays the same Quarto document "hello.qmd" in a plain text source editor.
- Content Below:** Shows the rendered content of the document, including the "Meet Quarto" and "Meet the penguins" sections, the penguin illustration, and the ggplot code.

Source: [Quarto Documentation](#) by Posit



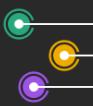
# Why Quarto, Not RMarkdown?

- a single ecosystem instead of multiple dependencies
- easier to organize/structure document and document layout
- better accessibility and richer features out of the box
- evaluate native language (R in knitr, Python/Julia in Jupyter)
- better ability to integrate multiple languages
- HTML slides with `revealjs` work in RStudio's visual editor



# *Communicating with Quarto*

## — Document Contents —



# Contents of a Quarto Document

**YAML Header**

**Markdown Text**

**Code Chunks**



# YAML Header

*Yet Another Markup Language...*

**YAML Ain't Markup Language!**

YAML is a human-readable data-serialization language.



# The YAML Header

All Quarto (and RMarkdown) files start with a YAML header.

```
1 ---
2 title: "My Quarto Document"
3 author: "Cedric Scherer"
4 format: html
5 editor: visual
6 ---
```



# The YAML Header

All Quarto (and RMarkdown) files start with a YAML header.

```
1 ---
2 title: "My Quarto Document"
3 author: "Cedric Scherer"
4 date: February 26, 2023
5 format:
6 html:
7 toc: true
8 toc-depth: 3
9 code-fold: show
10 knitr:
11 opts_chunk:
12 dev: "ragg_png"
13 retina: 1
14 dpi: 120
15 execute:
16 warning: false
17 message: false
18 fig-width: 10
19 fig-height: 6
```



# The YAML Header

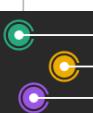
Fundamentally, a file written in YAML consists of a set of key-value pairs:

```
1 ---
2 key: value
3 ---
```

```
1 ---
2 format: html
3 ---
```

```
1 ---
2 format: pdf
3 ---
```

```
1 ---
2 format: revealjs
3 ---
```



# The YAML Header

Sub-options should be below the main format output and spacing matters:

```
1 ---
2 key:
3 value:
4 option1: text
5 option2: logical
6 ---
```

```
1 ---
2 format:
3 html:
4 toc: true
5 code-fold: true
6 ---
```



# The YAML Header

YAML is sensitive, so be careful:

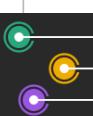
```
1 ---
2 format:html ## invalid, no space between
3 ---
```

```
1 ---
2 format: ## invalid, read as missing
3 html
4 ---
```

```
1 ---
2 format:
3 html: ## valid but needs next object
4 ---
```

Valid YAML can look a bit differently based on what all is needed

```
1 format: html ## valid, on a single line with a space
```



```
1 format:
2 html ## valid, indented and no trailing :
```

```
1 format:
2 html: ## valid, : needed for suboptions
3 toc: true ## valid, indented and no trailing :
```



# Markdown Text

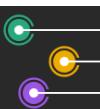
The plain-language paragraphs are added as simple text with Markdown formatting options.

Markdown is a simple markup language which allows to format text with a few defined symbols, including font styles, section headers, hyperlinks, and embedding of images.



# Markdown Text

- Text wrapped into
  - \* or \_ is formatted as *italic*
  - \*\* or \_\_ is formatted as **bold**
  - ~~ is formatted as ~~strikethrough~~
  - ` is styled as **verbatim code**
  - ^ creates a super<sup>script</sup>
- Hashes define the header level
  - # adds a level-1 header
  - ## adds a level-2 header (and so on)
- Different types of brackets can be used to
  - turn an URL into a hyperlink: <url>
  - turn an URL into a named hyperlink: [name](url)
  - include an image: ![caption](url\_or\_path)



# Markdown Text

- Using \*, + and -, or 1., i), and A. you can create lists.

## Unordered lists:

```
1 * item A
2 * item B
3 * subitem B1
4 * item C
```

- item A
- item B
  - subitem B1
- item C

## Ordered lists:

```
1 1. point 1
2 1. point 2
3 i) point 2.1
4 1. point 3
```

1. point 1
2. point 2
  - i. point 2.1
3. point 3



# Markdown Text

- Using \*, + and -, or 1., i), and A. you can create lists.

## Unordered lists:

```
1 - item A
2 - item B
3 + subitem B1
4 - item C
```

## Ordered lists:

```
1 2. point 2
2 1. point 1
3 i) point 2.1
4 3. point 3
```

- item A
- item B
  - subitem B1
- item C

- 2. point 2
- 3. point 1
  - i. point 2.1
- 4. point 3



# Markdown Text

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the Quarto code, and the right pane shows the rendered output.

**Quarto Code (Left):**

```
24 - ## Quarto
25
26 Quarto enables you to weave together content and executable code into a
finished document. To learn more about Quarto see <https://quarto.org>.
27
28 - ## Writing Markdown
29
30 Markdown is a simple markup language, which means that we can format text
with a few defined symbols.
31
32 The two hash tags `##` above define a second-level header (`

` in HTML[^1]), the back ticks used in the previous sentence format the text between those ticks as code. 33 34 [^1]: The markdown syntax `## Writing Markdown` actually gets rendered to proper HTML code as `Writing Markdown`. Ah, and btw: this is how you add footnotes. 35 36 37 - ### More Format Options 38 39 Text wrapped into a set of asterisks or underscores is format as *italic* text, while wrapping it into a double-asterisk or -underscore produces **bold** text. Of course, **we can** *can* those and also ***combine both***.


```

**RStudio Status Bar:** 96:58 | Chunk 4: my-ggiraph | Quarto

**Console:** Console

**Rendered Output (Right):**

**Quarto**

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

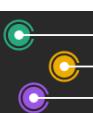
## Writing Markdown

Markdown is a simple markup language, which means that we can format text with a few defined symbols.

The two hash tags **##** above define a second-level header (**`h2`** in HTML<sup>[^1]</sup>), the back ticks used in the previous sentence format the text between those ticks as code.

### More Format Options

Text wrapped into a set of asterisks or underscores is format as ***\*italic\**** text, while wrapping it into a double-asterisk or -underscore produces ****bold**** text. Of course, ****we can**** ***\*can\**** those and also ****\*\*\*combine both\*\*\*****.



# Markdown Text

The screenshot shows the Quarto IDE interface with two main panes. The left pane is titled "hello-quarto.qmd x" and displays the source code for a Quarto document. The right pane shows the rendered output of the same document.

**Left Pane (Source):**

- File: hello-quarto.qmd
- Buttons: Render on Save, Render
- Text:

```
Quarto

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

## Writing Markdown

Markdown is a simple markup language, which means that we can format text with a few defined symbols.

The two hash tags `##` above define a second-level header (`h2` in HTML<sup>1</sup>), the back ticks used in the previous sentence format the text between those ticks as code.

### More Format Options

Text wrapped into a set of asterisks or underscores is format as *italic* text, while wrapping it into a double-asterisk or -underscore produces **bold** text. Of course, we **can** *can* those and also **combine both**.

In the first paragraph we sue angle brackets, which renders the link inside `<` and `>` to a hyperlink. If you want to give the link another name, we can use a combination of brackets and parentheses like this: [click me](#).
- Console: Chunk 4: my-ggiraph

**Right Pane (Rendered Output):**

- Buttons: Environment, History, Connections, Git, Tutorial, Files, Plots, Packages, Help, Viewer, Presentation, Edit
- Text:

```
Quarto

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

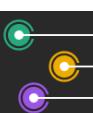
## Writing Markdown

Markdown is a simple markup language, which means that we can format text with a few defined symbols.

The two hash tags `##` above define a second-level header (`h2` in HTML<sup>1</sup>), the back ticks used in the previous sentence format the text between those ticks as code.

### More Format Options

Text wrapped into a set of asterisks or underscores is format as *italic* text, while wrapping it into a double-asterisk or -underscore produces **bold** text. Of course, we **can** *can* those and also **combine both**.



# Markdown Text

The screenshot shows the Quarto IDE interface with two main panes. The left pane is titled "hello-quarto.qmd" and is in "Source" mode, indicated by the selected tab. It contains the following content:

## Quarto

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

## Writing Markdown

Markdown is a simple markup language, which means that we can format text with a few defined symbols.

The two hash tags `##` above define a second-level header (`h2` in HTML<sup>1</sup>), the back ticks used in the previous sentence format the text between those ticks as code.

### More Format Options

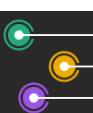
Text wrapped into a set of asterisks or underscores is format as *italic* text, while wrapping it into a double-asterisk or -underscore produces **bold** text. Of course, we **can** *can* those and also **combine both**.

In the first paragraph we sue angle brackets, which renders the link inside `<` and `>` to a hyperlink. If you want to give the link another name, we can use a combination of brackets and parentheses like this: [click me](#).

Chunk 4: my-ggiraph

Console

The right pane is in "Visual" mode, showing the rendered version of the document. The rendered content is identical to the source content, displaying the headings, text, and links as they would appear in a final document.



# Code Chunks

To interpret code we place it in so-called **chunks**.

R code chunks are identified with **{r}**:

```
```{r}
"Hello Quarto!"
1 + 1
```
```



# Code Chunks

... with optional chunk options in YAML style identified by # | at the begin:

```
```{r}
#| label: a-label-for-my-chunk
#| echo: false
#| warning: false
"Hello Quarto!"
1 + 1
```

```

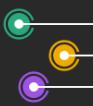
```
```{r}
#| label: a-ggplot-for-my-report
#| echo: false
#| fig-width: 15
#| fig-height: 10
ggplot(mpg, aes(x = cty, y = hwy)) + geom_point()
```

```



# *Communicating with Quarto*

## *— Rendering Documents —*



# The Rendering Process

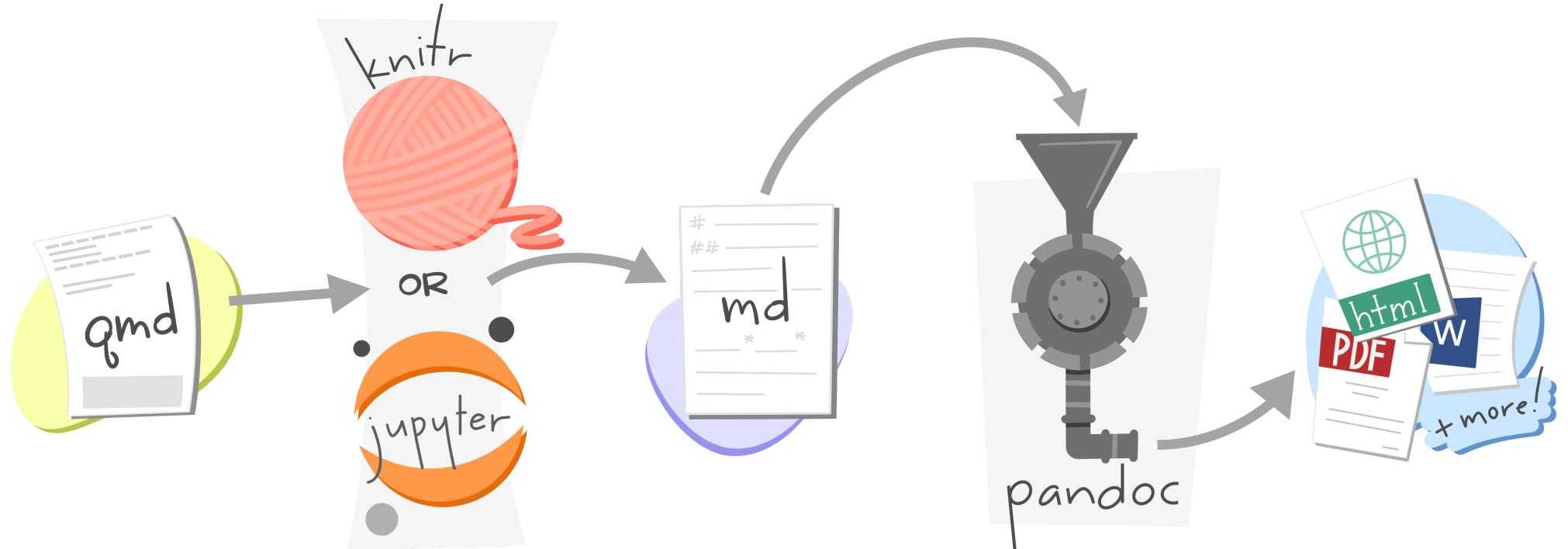
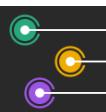
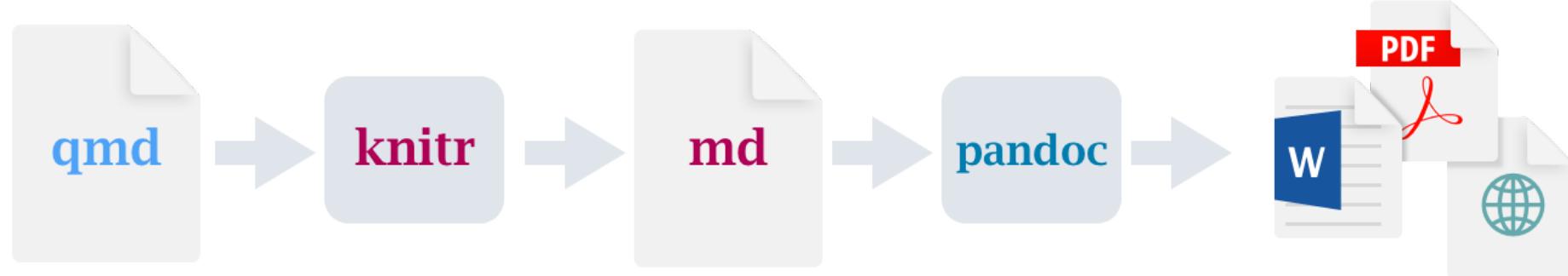


Illustration by [Allison Horst](#)



# The Rendering Process



Source: [Quarto Documentation](#) by Posit



Source: [R Markdown Documentation](#) by Posit



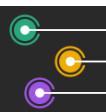
# Rmarkdown

TEXT. CODE. OUTPUT.  
(GET IT TOGETHER, PEOPLE.)



Illustration by [Allison Horst](#)

Cédric Scherer // R Course TU Dresden // Communicating with Quarto



# Rendering

The **{quarto}** R package provides some utilities when working with Quarto documents.

```
1 # install.packages("quarto")
2 quarto::quarto_render("document.qmd") ## defaults to html
```

```
1 quarto::quarto_render("document.qmd", output_format = "pdf")
```



# Rendering in RStudio

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the source code for "hello-quarto.qmd". The right pane shows the rendered output.

**Source:**

```
1 ---
2 title: "My Quarto Document"
3 author: "Cedric Scherer"
4 date: `r format(Sys.time(), '%B %d, %Y')`
5 format:
6 html:
7 toc: true
8 toc-depth: 3
9 code-fold: show
10 knitr:
11 opts_chunk:
```

**Quarto Context Menu:**

- Quarto
- Writing Mark...
- More Format ...
- Running Code
- Plots
- Tables
- Final Note

**RStudio Environment:**

- Environment
- History
- Connections
- Git
- Tutor

**Viewer Tab:**

- Files
- Plots
- Packages
- Help
- Viewer

**Output Preview:**

## My Quarto Document

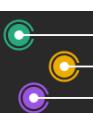
AUTHOR PUBLISHED  
Cedric Scherer February 25, 2023

## Quarto

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

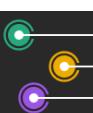
Output created: hello-quarto.html

Watching files for changes



# Rendering in RStudio

The screenshot illustrates the RStudio interface during the rendering of a Quarto document. The top navigation bar includes tabs for Environment, History, Connections, Git, and Tutorial, along with sub-tabs for Files, Plots, Packages, Help, and Viewer. The main workspace shows the Quarto configuration file `hello-quarto.qmd` in Source mode, with the `Render` button highlighted. A dropdown menu from the Render button lists options: Quarto, Writing Markd..., More Format ..., Running Code, Plots, Tables, and Final Note. To the right, the Viewer pane displays the rendered content: "My Quarto Document" by Cedric Scherer, published on February 25, 2023. Below the viewer, the Quarto documentation is shown, stating: "Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>". The bottom left pane shows the output of the rendering process: "Output created: hello-quarto.html" and "Watching files for changes".



# Rendering in RStudio

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the Quarto configuration file (hello-quarto.qmd) with the following content:

```
1 ---
2 title: "My Quarto Document"
3 author: "Cedric Scherer"
4 date: "`r format(Sys.time(),
5 format:
6 html:
7 toc: true
8 toc-depth: 3
9 code-fold: show
10 knitr:
11 opts_chunk:
```

The right pane shows the Quarto rendering context with the following details:

- Environment: History, Connections, Git, Tutorial
- Files, Plots, Packages, Help, Viewer
- Preview in Window (No Preview)
- Preview Images and Equations
- Show Previews Inline
- Chunk Output Inline (Chunk Output in Console)
- Expand All Output
- Collapse All Output
- Clear Output
- Clear All Output

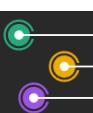
The preview pane displays the rendered Quarto document:

# My Quarto Document

| AUTHOR         | PUBLISHED         |
|----------------|-------------------|
| Cedric Scherer | February 25, 2023 |

## Quarto

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



# Rendering in RStudio

The screenshot illustrates the Quarto rendering workflow within RStudio. On the left, the Quarto document 'hello-quarto.qmd' is open, showing YAML front matter and code blocks. A purple box highlights the 'Render on Save' checkbox in the toolbar. The 'Quarto' dropdown menu is open, showing options like 'Writing Markd...', 'More Format ...', and 'Final Note'. The 'Console' tab shows the command '# My Quarto Document' and the output 'Output created: hello-quarto.html'. The 'Preview' tab shows the rendered HTML content.

hello-quarto.qmd x

Render on Save

Source Visual

1 ---  
2 title: "My Quarto Document"  
3 author: "Cedric Scherer"  
4 date: `r format(Sys.time(), '%B %d, %Y')`  
5 format:  
6 html:  
7 toc: true  
8 toc-depth: 3  
9 code-fold: show  
10 knitr:  
11 opts\_chunk:

21:4 # My Quarto Document Quarto

Console Terminal x Background Jobs x

Preview: hello-quarto.qmd Running 2:35

Output created: hello-quarto.html

Watching files for changes

Environment History Connections Git Tutorial

Files Plots Packages Help Viewer

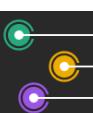
x Edit

My Quarto Document

AUTHOR PUBLISHED  
Cedric Scherer February 25, 2023

Quarto

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



# Navigating in RStudio

The screenshot shows the RStudio interface with the Quarto extension active. The left pane displays the source code for a Quarto document named "hello-quarto.qmd". The right pane shows the rendered output, which includes a title section, author information, and a summary of the document's purpose.

**Source View:**

```
1 ---
2 title: "My Quarto Document"
3 author: "Cedric Scherer"
4 date: `r format(Sys.time(), '%B %d, %Y')`
5 format:
6 html:
7 toc: true
8 toc-depth: 3
9 code-fold: show
10 knitr:
11 opts_chunk:
```

**Console View:**

Output created: hello-quarto.html

Watching files for changes

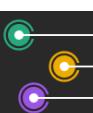
**Quarto Output:**

## My Quarto Document

| AUTHOR         | PUBLISHED         |
|----------------|-------------------|
| Cedric Scherer | February 25, 2023 |

## Quarto

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



# Navigating in RStudio

The screenshot shows the RStudio interface with the Quarto extension active. The left pane displays the R Markdown source code for a file named "hello-quarto.qmd". A context menu is open over the first few lines of code, with the "My Quarto Document" option highlighted. The right pane shows the generated Quarto document output, which includes the title "My Quarto Document", author information ("Cedric Scherer" published on "February 25, 2023"), and a section titled "Quarto" with a descriptive paragraph about its capabilities.

hello-quarto.qmd

Source Visual

1 ---  
2 My Quarto Document  
3 Quarto  
4 Writing Markdown  
5 More Format Options  
6 Running Code  
7 Chunk 1  
8 Chunk 2  
9 Plots  
10 Chunk 3: my-ggplot  
11 Chunk 4: my-ggiraph  
12 Chunk 5: my-ggplotly  
13 Tables

21:4 # My Quarto Document ▾

Console Terminal × Background Jobs ×

Preview: hello-quarto.qmd

Output created: hello-quarto.html

Watching files for changes

Environment History Connections Git Tutorial

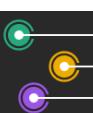
Files Plots Packages Help Viewer

My Quarto Document

AUTHOR PUBLISHED  
Cedric Scherer February 25, 2023

Quarto

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



# Your Turn: Hello Quarto

- Use RStudio to create a new default Quarto document.
- Explore the visual and source editor, and pick your favorite.
- Add a title, your name as the author, as well as the current date and a 2-level toc.
- Create two sections, one with things you learned this week and one with things you want to explore further.
- Create a third section with an image a plot of your choice.
- Create a fourth section with a ggplot of your choice.
- Hide the code used to create the visualization.
- Adjust the aspect ratio of the plot and increase its resolution.
- **Bonus:** Change the html theme to `cosmo` (or a theme of your choice)
- Render the report.
- **Bonus:** Explore options to create tables and interactive plots.

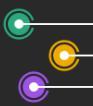


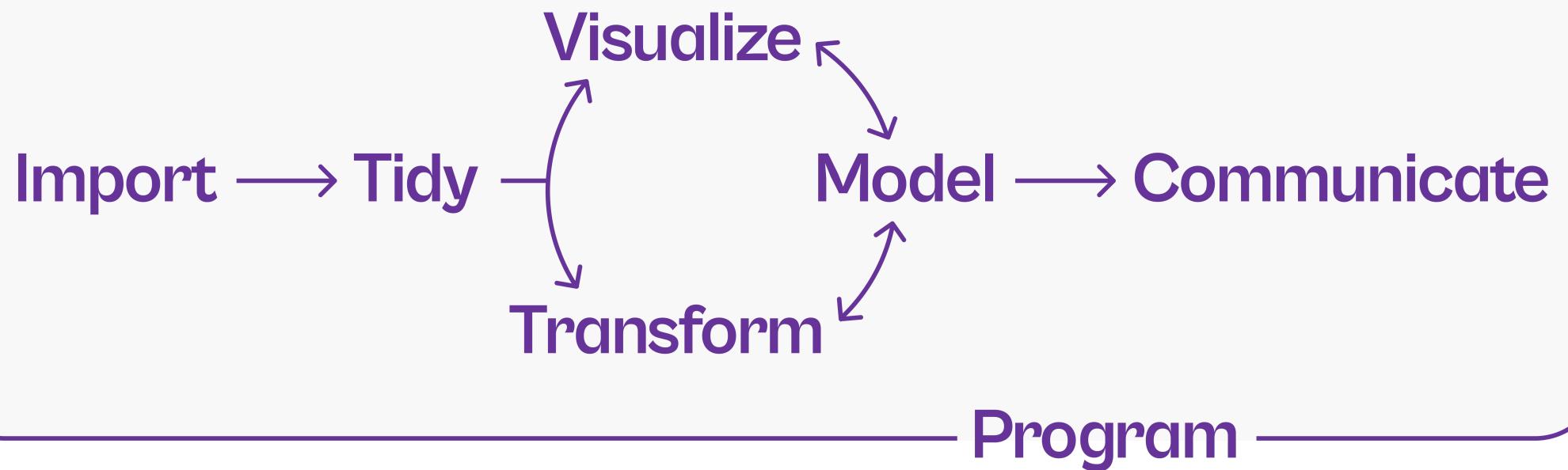
# Resources

- Quarto Documentation
- Quarto FAQ
- Quarto Discussions
- Extensive Workshop on Quarto
- The Quarto Keynote Talk at rstudio::conf(2022)

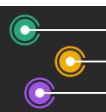


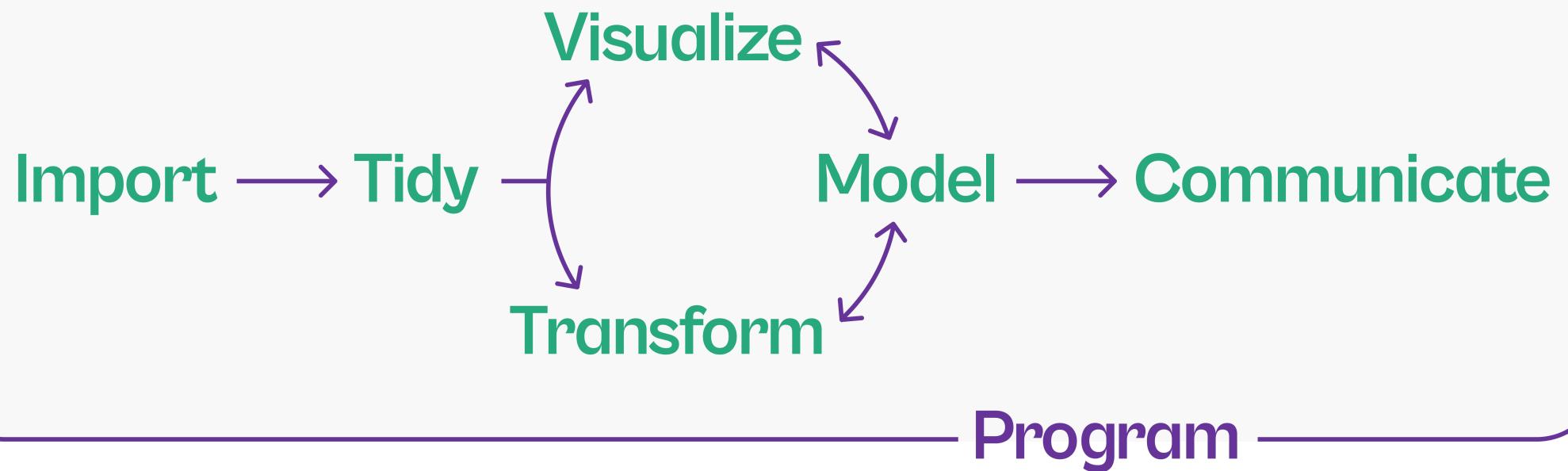
# *What's Next?*



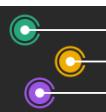


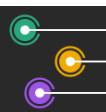
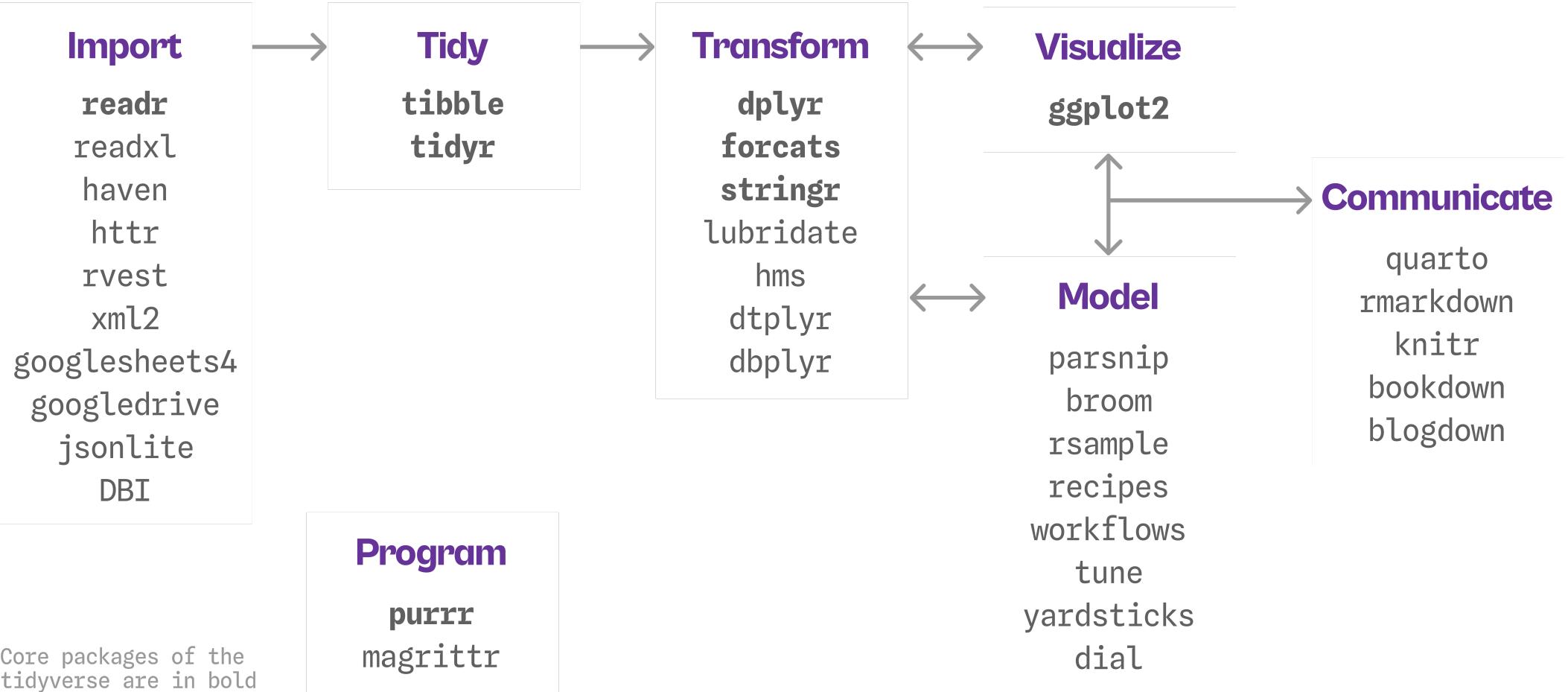
The data science workflow, modified from "[R for Data Science](#)"

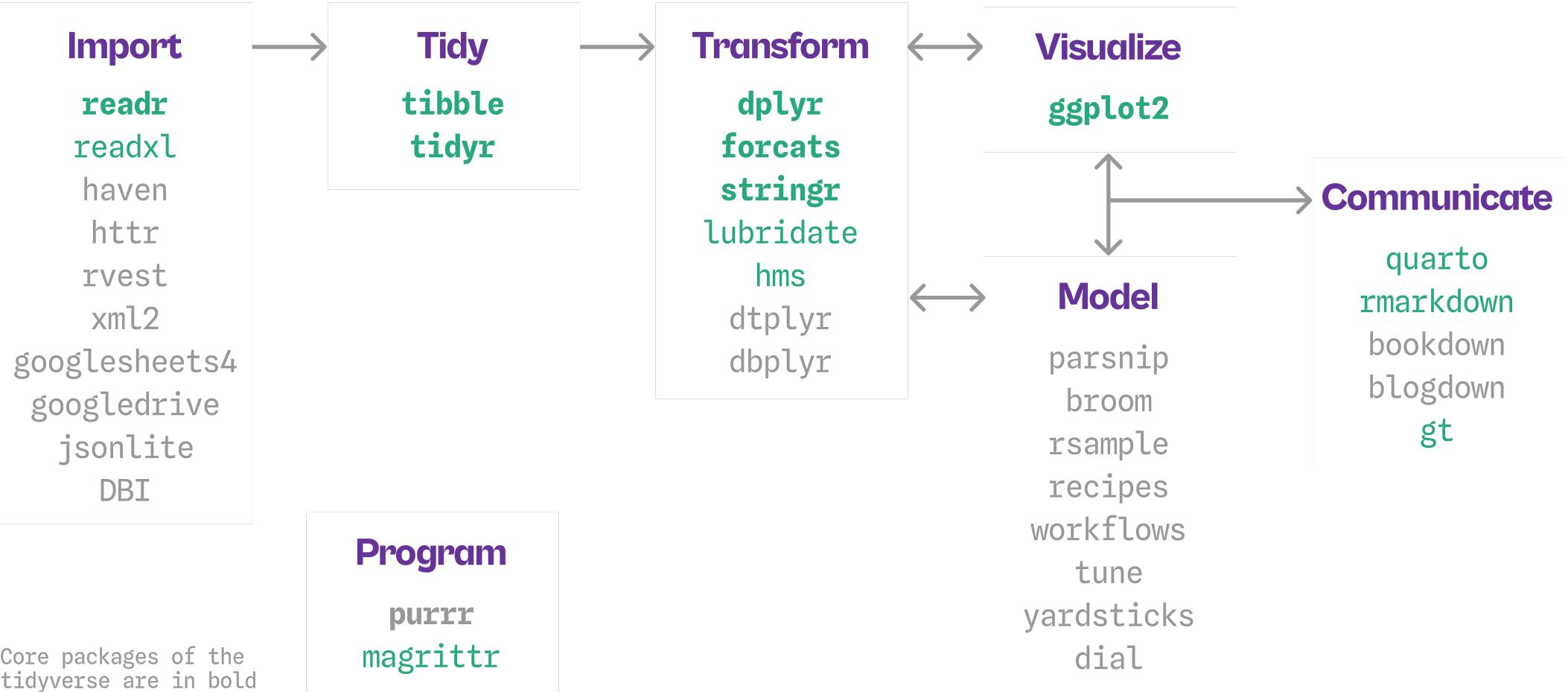




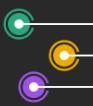
The data science workflow, modified from "R for Data Science"







# So... What's Next? — AMA // WOYO // CIAD —



# That's it Folks... — Thank you! —

