Table of contents

xercises: Introduction to Quarto	1
Overview	1
Set up	2
Exercise 1: Rendering	2
Tasks	2
Exercise 2: YAML metadata	2
Tasks	3
Exercise 3: Document body	3
Tasks	3
Exercise 4: Code chunks	4
Tasks	4
Exercise 5: Your turn!	4
Tasks	4

Exercises: Introduction to Quarto

Overview

These exercises introduce you to key elements of Quarto documents. You will have the opportunity to explore how these elements enable you to create quality documents that effectively communicate and document your analyses and findings.

There are five exercises to enhance your familiarity with Quarto documents. The first exercise introduces you to options for **rendering** a Quarto file into a document. The second exercise focuses on understanding **YAML headers**, where you will learn to customize metadata such as title and author. Next, you will dive into the **document body**, exploring formatting options to organize your content effectively. During the fourth exercise, you will be introduced to **code chunks**, allowing you to embed and execute code in R, Python, or Julia, creating dynamic reports that showcase your analysis. The final exercise provides an opportunity to apply your new skills by creating a Quarto report. Through these exercises, you will develop essential skills to create professional Quarto documents.

For additional resources on Quarto documents, please visit:

- Quarto references
- Quarto Guidance webpage
- Quarto Cheatsheet

Set up

• Complete the Prework steps to download the exercise materials

Exercise 1: Rendering

Rendering in Quarto documents refers to the process of converting a Quarto file, which is written in a markdown-like syntax, into a final output format such as HTML, PDF, or Word. During rendering, the Quarto system interprets the markdown content, processes embedded R code chunks, and generates the respective output by applying formatting specified in the YAML header. This transformation allows users to produce well-structured and visually appealing documents that incorporate text, data visualizations, and analysis results seamlessly, making it an essential step in creating dynamic reports and presentations.

Tasks

- \Box Open 1_rendering.qmd. What are three main components of the file?
- \Box Render the document using the RStudio GUI (\Longrightarrow Render).
- \Box Render the document using the Quarto CLI.

– Hint

In the terminal: quarto render 1_rendering/<input file name>

- \Box Render the document using the quarto R package.
 - Hint

In the console: quarto::quarto_render("./1_rendering/<input file name>")

Exercise 2: YAML metadata

A YAML (Yet Another Markup Language) header is a crucial component of Quarto documents that defines metadata for the document, typically enclosed within a pair of triple-dashed lines at the beginning of the file. It allows users to specify important information such as the document's title, author, date, and output formats (like HTML, PDF, or Word). Additionally, YAML headers can include parameters to control formatting options and customize the behavior of the document, such as including bibliographies or setting specific themes. By using a YAML header, authors can ensure that their documents are well-organized and professionally presented.

Tasks

- □ Open 2_yaml_metadata.qmd and render the document.
- \Box Update the subtitle and replace the author name with your name. Re-render it.
- \Box Change the output to a PDF and render it. What did you notice?
- \Box Now render it as a MS Word document (docx). Open the file to view the results.

Bonus tasks

 \Box Render the document as a Word document using the MS_Style.docx file in the templates folder.

- **Hint** Using templates

 \Box Render the document simultaneously as a Word document and a PDF.

– Hint

Add output_format = "all" argument to quarto::quarto_render()

Exercise 3: Document body

The document body of a Quarto document is the main content area where the narrative, analysis, and findings are presented. This section is typically written in markdown format, allowing users to easily incorporate various elements such as headings, paragraphs, lists, tables, and images. The importance of the document body lies in its role as the primary medium for communicating information and insights to the reader; it organizes thoughts and results in a coherent and engaging manner. A well-structured document body enhances readability and comprehension, enabling the audience to grasp complex ideas and data more effectively. By integrating both text and visual elements, the document body plays a critical role in making the overall document informative and visually appealing.

Tasks

Open 3-document_body.qmd and make the following changes to the document:

Inline:

- \Box Italicize the species name.
- \Box Make "western snowy plover" into a hyperlink to the FWS page or Wikipedia page for the species.

Block:

- \Box Create a new header named "My List".
- \Box Make an ordered and an unordered list.
- □ Insert an image of the western snow plover below the "U.S. Fish and Wildlife Service" header. The image can be found in the ./images folder.

Exercise 4: Code chunks

Code chunks in a Quarto document are sections of embedded code, typically written in R or other programming languages, that allow users to perform data analysis, generate visualizations, and execute computations directly within the document. These chunks are enclosed by specific delimiters, enabling Quarto to recognize and execute the code during the rendering process. The importance of code chunks lies in their ability to integrate dynamic content with narrative text, making it possible to showcase real-time analysis and results alongside explanatory writing. This functionality enhances reproducibility, as the document can simultaneously present the code and its output, facilitating transparency and allowing others to verify or replicate the findings. By incorporating code chunks, Quarto documents become interactive and informative, transforming static reports into live analytical narratives.

Tasks

Open 4_code_chunks.qmd and make the following changes to the rendered document:

- □ Label the plot code chunk to help you differentiate from other chunks.
- □ Update the figure caption to read "Mean monthly Western..."
- \Box Add + theme_classic() to the end of the ggplot script to clean up the plot. Re-render.
- □ Remove the output of a code chunk. Change the include option to false and re-render.

Exercise 5: Your turn!

You have now learned how to render a Quarto document and how to change the look and feel of a Quarto document by customizing the YAML metadata, inline and block elements of the text body, and code chunks. Now it is time to apply these skills by creating a Quarto document from scratch.

Tasks

Do your best to recreate the document below. When rendered as a .html, your file should look exactly the same aside from the publishing date.

Plover Report

PUBLISHED May 20, 2025

Introduction

Background

Plovers are cute.

Methods

1. Find plovers.

2. Count plovers.

Results

In the 2025 survey, the plover index was 62.8318531. Many males were counted (Figure 1).



Histogram of plovers

Figure 1: Histogram of male plovers

Use the following steps to guide you:

 \Box Create a new Quarto document, and save it to your Exercise 5 folder.

Hint

File > New File > Quarto Document

- □ Edit the YAML to match the title and *today's* date. Any time your file renders, it should display the current day. Add a parameter called alpha, and set alpha to a value of 20.
- □ Write out your headers and fill in the report content. Don't forget to match styling (e.g., bold, italics, lists) where applicable.
- □ In the **Results** section, use inline R code for the survey year. You can use the following code to obtain the current year: format(Sys.Date(), '%Y')
- \Box Compute the plover index using inline R code. The formula for the imaginary plover index is $\pi \alpha$, where α is the parameter you defined in the YAML.
- □ Use a code block and the function hist() to generate a histogram of male plover counts. Make sure to add your figure caption and ensure that the code chunk does not display when you render. Below, you will find some helpful code to get the list of male counts, named plovers, from your data file:

```
data <- read.csv("data/plover_data.csv")
plovers <- data$Males</pre>
```

Hint

The chunk options you are looking for are echo and fig-cap. Use hist(plovers) to generate the plot.

Bonus tasks

- □ Replace "Many" in the **Results** section with the true total of male counts from the **plovers** list. Suggestion: use the **sum()** function.
- \Box Your supervisor tells you that the α value for your calculations is actually 30. Update and re-render your file without changing the document body. Your supervisor also tells you she wants you to send her a Word document of the report but upload a PDF to ServCat. Render both formats.