R Markdown & Quarto Guide

Table of contents

## 1 Introduction

**R Markdown** and **Quarto** are extremely helpful for creating reproducible work, documentation, or for creating meaningful reports. R Markdown was released in 2014 as a cornerstone of R-based reporting. Quarto is its successor, developed in 2022 to “make the process of creating and collaborating on scientific and technical documents dramatically better”.

## 2 What is R Markdown?

**R Markdown** is a document format introduced by RStudio/Posit that allows users to combine:

* Narrative text written in **Markdown** (you will see these files on most GitHub pages),
* **R code chunks** for analysis,
* Embedded **outputs** (e.g., plots, tables, models)

### 2.1 File Structure

An R Markdown document uses the .Rmd extension and typically consists of:

* [A **YAML** header](https://monashdatafluency.github.io/r-rep-res/yaml-header.html)
* Markdown-formatted narrative
* Code chunks enclosed in triple backticks and labeled with {r}. Note that you can actually run different languages in R Markdown using triple backticks and the label {python} for example.

### 2.2 Output Formats

With R Markdown, you can render documents to:

* HTML (via html\_document)
* PDF (via LaTeX and pdf\_document)
* Word (word\_document)
* Presentations (ioslides, slidy, xaringan)
* Dashboards (flexdashboard)
* Shiny apps (shiny)

## 3 What is Quarto?

**Quarto** is the new open-source scientific and technical publishing system designed as the successor to R Markdown. Quarto is designed to be **language-agnostic**, supporting R, Python, Julia, and JavaScript and create more advanced projects like **books**, **websites**, and **Shiny apps**.

### 3.1 File and Project Structure

Quarto files use the .qmd extension and have a structure similar to .Rmd, but with a bit different YAML.

## 4 Similarities

| Feature | RMarkdown | Quarto |
| --- | --- | --- |
| Code+TextIntegration | Yes | Yes |
| Markdown-basedSyntax | Yes | Yes |
| HTML/PDF/WordOutputs | Yes | Yes |
| ShinySupport | Yes | Yes(betterinQuarto) |
| Citations | Yes | Yes |
| RStudio/VSCodeSupport | Yes(RStudio) | Yes(RStudio+VSCode+Jupytr) |
| PlotOutput,Tables | Yes | Yes |

## 5 Differences

### 5.1 Language Support

| Feature | RMarkdown | Quarto |
| --- | --- | --- |
| R | Yes | Yes |
| Python | Yes(viareticulate) | Yes(native) |
| Julia | No! | Yes |
| ObservableJS | No! | Yes |

R Markdown supports only R (or Python via reticulate, with limitations). *Note that you can also use SQL in R Markdown (I have used this for a project)*. Also **Quarto** supports native **websites**, **blogs**, and **books**, whereas R Markdown requires external packages.

## 6 Example

In R Markdown and Quarto we use code chunks. These code chunks can have many options. Below are the common code chunks.

| Option | What it does |
| --- | --- |
| echo=FALSE | Hides the code from output but still runs it. Useful for setup or styling. |
| eval=FALSE | Shows the code, but doesn’t run it. Use when you want to show an example. |
| include=FALSE | Runs the code but shows neither the code nor the output. |
| message=FALSE | Suppresses messages |
| warning=FALSE | Hides warnings generated by the code. |
| fig.cap="..." | Adds a caption to a plot or image. |

### 6.1 Different Languages

Below we list the numbers from 1 to 10 using R

for (i in 1:10){  
 print(i)  
}

[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5  
[1] 6  
[1] 7  
[1] 8  
[1] 9  
[1] 10

Below we list the numbers from 1 to 10 using Python

for i in range(1,11):  
 print(i)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

Below we list the numbers from 1 to 10 using JavaScript

Array.from({length: 10}, (\_, i) => i + 1)