Overview of Quarto in RStudio Environment

Part of Workshop Entitled "Introduction to Quarto: Quarto's Capabilities for Work and School Planning Board"

Jae Min Jung, Ph.D.

2/02/2025

Table of contents

R and RStudio

- Brief History
- Installation
    1. R: https://www.r-project.org/
    2. RStudio:
        - Desktop https://posit.co/download/rstudio-desktop/
        - Posit Cloud: https://posit.cloud/

. . .

- R Studio IDE menu (Next slide)

- Three major file types in RStudio (Next slide)

Demonstrate

RStudio IDE

- Four quadrants
- Menu
  - Tools > Global Options
  - Output location options
    - Manually set `working directory`
    - Automatic method with the `Rproj` file (Next slide)

Steps in Preparing for a project

1. Start a new project

- Projects are the containers for all of your notebooks.

```
1.  From File, click `new project`
2.  Select a new or existing directory depending on your needs
3.  Select your folder that will contain your project.
4.  Press create project
```

- You should see your `.Rproj` file and others in the `Files` panel

2. Start a codebook/notebook and save it

- R Scripts file
- Rmd file
- Quarto file

Markdown vs. HTML

Markdown and HTML are both markup languages used for creating formatted content, but they have distinct characteristics. Here's a summary of their similarities and differences:

Feature	Markdown	HTML
Syntax complexity	Simple and easy to read	More complex with specific tags
Learning curve	Low, quick to learn	Steeper, requires more time to master
Readability	Highly readable, even in raw form	Less readable in raw form
Flexibility	Limited formatting options	Highly flexible with extensive formatting
Output	Primarily static content	Dynamic web pages and applications
Supported elements	Basic formatting (headings, lists, links)	Wide range of elements (forms, multimedia)
Conversion	Easily converts to HTML	Cannot be directly converted to Markdown
Collaborative editing	Well-suited for collaboration	Less ideal for collaborative editing
Customization	Limited styling options	Extensive styling with CSS
Use cases	Documentation, simple content	Complex web development, detailed layouts
Browser support	Requires conversion to HTML	Natively supported by browsers
Extensibility	Limited, varies by flavor	Highly extensible with JavaScript

3. Start literate coding

What is Quarto?

- Quarto is an open-source technical publishing system designed for creating a wide variety of documents and publications (https://quarto.org/)

. . .

- Support multiple coding languages:
  - R, Python, Julia, ObservableJS
- Supported by multiple IDEs

  - RStdudio, VS Code, Jupyter, Neovim

. . .

- Publish to multiple hosting services
  - QaurtoPub, GitHub Pages, Posit Connect, Netlify, Hugging Face, etc.

. . .

- Great for **reproducible research/publications/documents**.

Quarto Work Flow Basics (Demo)

- How to start it and save it (e.g., test.qmd)
- Rendering it:
- Source vs. visual tab interface
- r4ds: https://r4ds.hadley.nz/quarto#quarto-basics

demonstrate it

Three Areas in Quarto File

- Detailed tutorials available at the official Quarto site

---

1. Yaml header

An YAML header demarcated by three dashes (`---`) on either end.

```
---
title: "Module 1 Introduction to R, RStudio, and Qurto"
author: "Jae Jung"
date: '2025-02-02 19:26:18'
format:
  html:
    toc: true
    toc-depth: 4
    embed-resources: true
editor: visual
execute:
```

```
  freeze: auto
---
```

---

2. Code chunk

```
```{r}
#| label: demo-code-chunk
#| include: true

#install.packages("tidyverse")
#install.packages("palmerpenguins")
library(tidyverse)
library(palmerpenguins)
```
```

---

3. Markdown text area

- Text area is all the canvas area within qmd file other than Yaml header and code chun areas.
- Quarto uses markdown syntax for text.
- You can use text area for typing prose as you would normally do in MS Word or Google Doc.
  - Text with formatting: section headers, hyperlinks, an embedded image, and an inline code chunk.
- You can also style it: e.g., **Bold**; *Italicize*

Coding Tips

Be Careful: do **not** ***code*** in the **text area**.

- It is possible to type the code and run in the text area.

- However, it won't be read and rendered into a document.
- During the rendering, RStudio will be in an auto-pilot mode and will treat everything in the text area as a text except for `in-line code`.

Qurto Interface

Visual editor

- Easier to those who are familiar with MS Word or Google Doc.
- To add something; press `ctrl/commnad` + `/`
- Adding table by hand is cumbersome.
- Adding an figure/image

Source editor

- Easier to those who are familiar with R Script file or Rmd file.
- Useful for debugging any Quarto syntax errors since it's often easier to catch these in plain text.
- Handy reference sheet available at the RStudio menu:
  - *Help* > *Markdown Quick Reference*

Literate Programming in Quarto

Basic Operations

In the following code chunk, you will learn some basic operations in R.

```
1+1
2*2 # *: multiplication
2^3 # ^: use carrot to raise the base to the power of the following number.

# creating an object
message <- "Hello WOrld!"

message = "Hello WOrld!" # equal sign also works.
```

```r
# to print,
print(message)

# to print, print function is necessary. You can just type the object and run it.
message

# An object can be any type: e.g., strings and numbers
number <- 7

max(2,5,90,30) # maximum

min(2,5,90,30) # minimum
```

````
```{r}
#| label: Basic-Operation
1+1
2*2 # *: multiplication
2^3 # ^: use carrot to raise the base to the power of the following number.

# creating an object
message <- "Hello WOrld!"

message = "Hello WOrld!" # equal sign also works.

# to print,
print(message)

# to print, print function is necessary. You can just type the object and run it.
message

# An object can be any type: e.g., strings and numbers
number <- 7

max(2,5,90,30) # maximum

min(2,5,90,30) # minimum
```
````

Coding Styles

```
#install.packages("tidyverse")
#install.packages("palmerpenguins")
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(palmerpenguins)
```

Base R way of coding

```
head(penguins)
```

```
# A tibble: 6 x 8
  species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
  <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
1 Adelie  Torgersen           39.1          18.7               181        3750
2 Adelie  Torgersen           39.5          17.4               186        3800
3 Adelie  Torgersen           40.3          18                 195        3250
4 Adelie  Torgersen           NA            NA                  NA          NA
5 Adelie  Torgersen           36.7          19.3               193        3450
6 Adelie  Torgersen           39.3          20.6               190        3650
# i 2 more variables: sex <fct>, year <int>
```

```
mean(penguins$bill_length_mm, na.rm = TRUE)
```

```
[1] 43.92193
```

Tidyverse way of coding

```
penguins |>
  head()
```

```
# A tibble: 6 x 8
  species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
  <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
1 Adelie  Torgersen           39.1          18.7               181        3750
2 Adelie  Torgersen           39.5          17.4               186        3800
3 Adelie  Torgersen           40.3          18                 195        3250
4 Adelie  Torgersen           NA            NA                  NA          NA
5 Adelie  Torgersen           36.7          19.3               193        3450
6 Adelie  Torgersen           39.3          20.6               190        3650
# i 2 more variables: sex <fct>, year <int>
```

```
penguins |>
  pull(bill_length_mm) |>
  mean(na.rm = TRUE)
```

```
[1] 43.92193
```

Pipe Operator

> Tip
>
> - R has multiple ways to accomplish the same goal.
> - `|>` is called `native pipe operator`. It works the same as `%>%`, which came from `mgrittr` package that revolutionized the way we code in R, paving the trend for the modern data science in R.
> - The pipe operator is one important difference between `base R` and `Tidyverse` in how we code.

Quarto Document Types

HTML

```
title: "Testing for Word document"
format: html
editor: visual
---
```

PDF

- In order to create PDFs you will need to install a recent distribution of LaTeX.
- Use TinyTeX (which is based on TexLive), which you can install with the following command:

```
Terminal
```

```
quarto install tinytex
```

```
---
title: "Testing for Word document"
format: pdf
editor: visual
---
```

MS Word

```
---
title: "Testing for Word document"
format: docx
editor: visual
---
```

Revealjs Presentation

```
---
title: "Testing for Presentation"
format: revealjs
editor: visual
---
```

Dashboard

```
---
title: "Testing for Presentation"
format: dashboard
editor: visual
---
```

Multiple Formats Option

```
---
title: "Housing Prices"
author: "YOur Name"
highlight-style: pygments
format:
  html:
    code-fold: true
    html-math-method: katex
  pdf:
    geometry:
      - top=30mm
      - left=30mm
  docx: default
---
```