

Lab 01

Setting up Julia, GitHub, and Quarto

CEVE 421/521

Fri., Jan. 12

Overview

Labs are in-class exercises intended to get practice with coding or analysis workflows.

- Instructions available on website
- Download ahead of time by using link from Canvas
- You will have your own repository (more in a minute)
- Try to finish in class, but due in 1 week

In this class, we will use

1. Julia
2. GitHub
3. Quarto
4. VS Code

Julia

- Syntax
 - Readable to computers and humans
 - Closely parallels math notation
- Designed for numerical and scientific computing
- Fast!
 - “Two language problem” (all you need is Julia)
 - Write intuitively
 - Performant libraries are written in Julia (easier installation)
- Open source

GitHub

1. You need a GitHub account
2. Code is stored in “repositories”
3. `clone` a repository to your computer
4. Make changes and `commit` them
5. `push` your changes to GitHub
6. Using GitHub classroom, instructors can view your code

Quarto

Quarto is a tool that allows you to combine text and code and create many types of output

- This website is made with Quarto
- You will use Quarto to create reports for labs
 - Everything in one place

- No running code, save a figure to **Downloads**, copy into Word, then update your code and try to remember where to paste it
- Reproducible

VS Code

1. VS Code is a text editor
 1. If you are an advanced user of another text editor, you can use that instead, but I recommend VS Code
2. VS Code can work as a Julia IDE

Getting set up

For detailed instructions, see the [Software Installation Guide](#).

Instructions

1. Read and go through the [Software Installation Guide](#) for instructions on setting up your computer for this course.
2. Follow the link to lab 1 assignment from Canvas (it should start with `classroom.github.com`). You will get a message saying "Your assignment repository has been created: ...". Click on the link to go to your repository.
3. **clone** the repository for lab 01 (use the Github Classroom link from Canvas) to your computer. You can use VS Code functionality, GitHub Desktop, or your terminal.
4. Open the directory containing the repository in VS Code doing one of the following:
 1. From GitHub desktop: **Repository > Open in Visual Studio Code**
 2. In VS Code: **File > Open Folder...**
5. Open the `index.qmd` file in VS Code and replace the `author: CEVE 421/521` line with your name and netID
6. Open the JULIA Repl
 1. Open the *command palette* (Ctrl+Shift+P on Windows/Linux, Cmd+Shift+P on Mac)
 2. Start typing "Julia: Start REPL". It will auto-complete; select the command as it appears.
7. Set up your project environment.
 1. In the Julia REPL, type `]` to enter the package manager. It should now show something like `(lab-01) pkg>`.
 2. Type **instantiate** and run it (Enter). This will install all the packages needed for this lab.
 3. Type the backspace key to exit the package manager.
8. Edit the `index.qmd` file to add your name and netID
9. Render the document
 1. Open the `solutions.qmd` file
 2. Open the *command palette* and run "Quarto: Preview". After some activity, a preview of the rendered document should open in VS Code. If you see something like **Browse at `http://localhost:4200/index.html`** you can open that link in your web browser to see the rendered document.
 3. Check the box on line 46 or 47 of the `solutions.qmd` file to indicate that you were able to render the document. If you were unable to render the document, check the other box and seek help. Make sure the box check renders correctly in the preview.
 4. Render to PDF or Microsoft Word following the directions below.
10. If you're still having trouble:
 1. Try running `build IJulia` in the Julia REPL's Pkg mode (type `]`)
 2. Come to office hours
 3. Post on the Canvas discussion for Lab 1
11. **commit** and **push** your changes

Running Code

We can use Quarto to run Julia code in-line

```
println("I'm using Julia!")
```

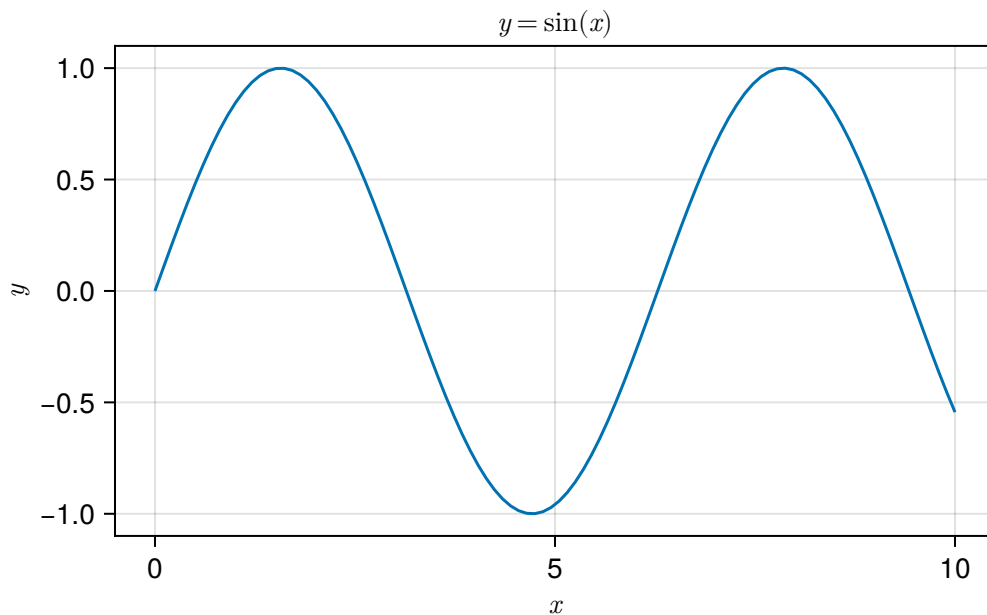
I'm using Julia!

We can also load packages

```
using CairoMakie
using LaTeXStrings
```

and use them to make plots

```
f = Figure()
ax = Axis(f[1, 1]; title=L"$y = \sin(x)$", xlabel=L"$x$", ylabel=L"$y$")
x = range(0, 10; length=100)
y = sin.(x)
lines!(ax, x, y)
f
```



Rendering the Document

A Quarto document is a plain text file that uses a simple code language to embed text, math, code, and the code's output. You can run the document line by line, or you can render to a variety of formats. This is helpful for sharing with others, or for creating a final document for yourself.

We will check our ability to

1. *Preview* the document in HTML (a web-native format)
2. *Render* the document to PDF or Microsoft Word (a portable, shareable, and printable format)

HTML

First, verify that you can preview the document in HTML:

1. Open the command palette (Cmd+Shift+P on macOS, Ctrl+Shift+P on Windows/Linux)
2. Type “Quarto: Preview”

PDF

You can use Quarto to generate PDF documents. Follow the instructions [on Quarto’s website](#) to install the necessary software.

1. To instruct Quarto to render to PDF, you need to edit the document metadata. Specifically, uncomment the lines in the document metadata corresponding to the PDF format.
2. Open the command palette (Cmd+Shift+P on macOS, Ctrl+Shift+P on Windows/Linux)
3. Type “Quarto: Render”

Word

You can also render the document to Microsoft Word:

1. To instruct Quarto to render to Word, you need to edit the document metadata. Specifically, uncomment the lines in the document metadata corresponding to the Word format.
2. Open the command palette (Cmd+Shift+P on macOS, Ctrl+Shift+P on Windows/Linux)
3. Type “Quarto: Render”

Feedback

Check the box (replace the space with an x) to confirm that this worked

- ☐ this worked for me
- ☐ this did not work for me (specify the error below)