Lab 1: Software Installation

Julia, Quarto, GitHub

CEVE 421/521

Fri., Jan. 12

1 Labs

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

2 Toolkit: overview

Getting set up for this course requires the following steps. If you are an experienced programmer, you are free to follow your own workflow to set up these tools. You will absolutely need Quarto, GitHub, and Julia. If you are not an experienced programmer, the following steps are not the *only* way to get these tools set up, but they are a *very good* way.

If you install course tools using steps other than the ones provided on this page, be aware that your instructors may be able to provide you with only limited support.

2.1 Julia

We're using the Julia programming language in this course. Julia is a fast, modern, open-source programming language designed for numerical and scientific computing.

We're using it for a few key reasons. First, the syntax is human-readable and closely parallels math notation, which reduces the cognitive burden of translating between conceptual and computational models. Second, it's fast, which means that Julia solves the "two language problem": you don't need to learn C or Fortran to dig under the hood and write fast code. Other great features include that it's open-source, which makes it reproducible and shareable, and it has a fantastic package manager.

There are some great resources out there about why Julia is great. See posts by Julia Data Science or the Julia Creators (with followup).

2.2 GitHub

`git` is a software tool for version control that keeps track of changes to files over time. GitHub is a website that hosts git repositories and provides a web interface for interacting with them.

To use GitHub, you'll need a GitHub account. Code on GitHub is stored in *repositories*. A simple workflow is to `clone` a repository to your computer, make changes, `commit` them, then `push` your changes to GitHub.

We will also use GitHub classroom, which allows instructors to share templates and view your code.

2.3 Quarto

Quarto is a tool that allows you to combine text and code and create many types of output. For example, this website is made with Quarto! You will use Quarto to create reports for labs. This lets you keep everything in one place – no more running code, saving a figure to `Downloads`, copying into Word, then trying to remember where to paste the figure when you update the code.

2.4 VS Code

VS Code is a text editor. If you are an advanced user of another text editor, you can use that instead. However, VS Code is very nearly an officially supported IDE for Julia.

3 Installing software

3.1 Install Julia

The best way to install Julia is through the `juliaup` tool, which will let you easily manage versions in the future and works seamlessly with VS Code. The instructions can be found at the JuliaUp GitHub repository, but we will summarize them here.

If your computer uses Windows, you can install Juliaup from the Windows Store.

If you have a Mac (or are using Linux), open a terminal (such as the Terminal app) and enter:

```
curl -fsSL https://install.julialang.org | sh
```

Once you install Juliaup, install Julia version 1.9 by opening a terminal (in MacOS or Linux) or the command line (in Windows) and entering:

```
julia add 1.10
julia default 1.10
```

This will install Julia 1.10 and make it the default version, which should maximize package compatibility throughout this course. Going forward, if you want to add new versions or change the default, you can follow the Juliaup instructions.

3.2 Install VS Code

VS Code is as close to an officially supported editor for Julia as you can get. We will follow this guide for setting up VS Code with Julia.

You can skip this section if you are an experienced programmer and already have a preferred IDE. Your IDE will likely have instructions for Julia and Quarto setup.

You can download VS Code here; open the downloaded file to install. Make sure to select the correct version for your operating system. If you have a newish Apple mac (with M1, M2, or M3 chip), make sure to check whether you have an Intel or Apple chip before choosing which version to download. You can also use homebrew or your preferred package manager to install VS Code.

3.2.1 VS Code Julia Extension

Like many IDEs, VS Code is a modular system that can be extended with plugins. We will install the Julia extension, which will allow us to run Julia code and interact with the Julia REPL from within VS Code (we'll add the Quarto extension later).

1. Open VS Code.
2. Select View and click Extensions to open the Extension View.
3. Search for `julia` in the search box. Click the green install button.
4. Restart VS Code once the installation is complete. It should automatically find your Julia installation; reach out if not.

The Julia VS Code extension offers you some nice features. You can start a REPL (an interactive Julia coding environment) by opening the "Command Palette" (View -> Command Palette, or CTRL/CMD+SHIFT+P) and typing "REPL" to bring up "Julia: Start REPL". You can also create `.jl` and `.qmd` files to write Julia code and execute line by line.

3.3 GitHub

See GitHub official tutorials for more helpful resources and tutorials.

3.3.1 Create GitHub Account

If you already have a GitHub account, you can use that for this course and do not need to create a new account. Otherwise, create an account. It doesn't have to be linked to your Rice email or your NetID.

For labs and projects, you should use the GitHub Classroom link posted on Canvas to "accept" the assignment, which will give you your own GitHub repository for that assignment. The first time you click one of these links, you will need to link your place on the course roster with your GitHub account.

3.3.2 GitHub Desktop (Optional)

You can do everything that you will need to do for this course with GitHub directly through VS Code. The GitHub desktop app is also great, or alternatively you may work directly through the terminal if you have prior experience.

3.4 GitHub Copilot (Optional)

GitHub Copilot is an AI-powered tool that helps you write code. You can install it following instructions here As described in the quickstart guide, Copilot is free for students. Be sure to review the policy on AI language models in the syllabus!

3.4.1 Install Git

`git` is a version control software that powers GitHub under the hood (`git` is the version control software, GitHub is an online platform). Based on past experience with the course, *you probably already have `git` installed!* If you're not sure if it's installed, see instructions here.

3.5 Install Quarto

Quarto combines the best of Jupyter notebooks and R Markdown to create a document format that is ideal for conducting and communicating data science. We will use Quarto to create and share our work in this course; this website is also built using Quarto.||

Follow the documentation to install Quarto. Be sure to ensure that you have the right version for your operating system.

3.5.1 Install the Quarto Extension for VS Code

Under "Step 2", click on the VS Code icon.

3.5.2 Install Jupyter

Under the hood, Quarto uses Jupyter to run code. You don't need to know how Jupyter works or worry about it, because it runs under the hood, but we will need to install it. Jupyter is a Python package.

If you don't have Python installed (if you're not sure, you should install Miniconda below), you'll need to install it. The best way, by far, is to install Miniconda (see Conda documentation).

Once you have Python installed, you can open your Terminal (open VS Code then open the terminal), then run

```
python3 -m pip install jupyter
```

> **!** Important
>
> Based on past experience, getting Jupyter installed and Quarto to find your Jupyter installation is the most common source of problems, especially on Windows machines. Please start this early so you have a full week to get help if you need it.

4 Lab Instructions

Once you have everything set up, your lab for the week will be to complete the following steps. These will ensure that your installation is working smoothly!

1. 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.
2. `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.
3. 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...`
4. Open the `index.qmd` file in VS Code and replace the `author: CEVE 421/521` line with your name and netID
5. 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.
6. 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.
7. Edit the `index.qmd` file to add your name and netID
8. 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.
9. 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
10. `commit` and `push` your changes to GitHub
11. Submit your rendered `.docx` or `.pdf` file to Canvas

4.1 Running Code

We can use Quarto to run Julia code in-line

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

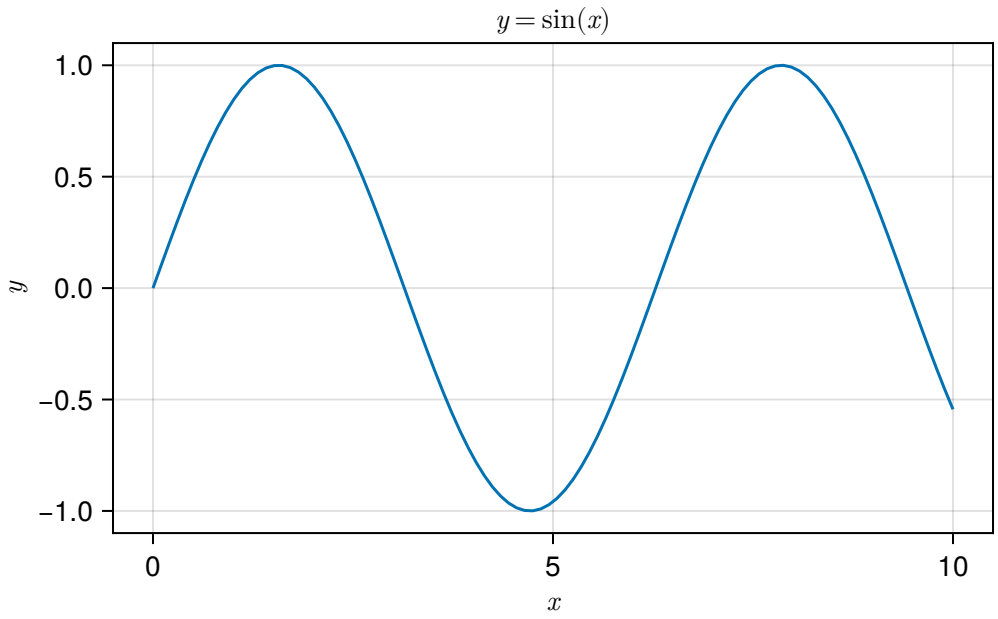
```
I'm using Julia!
```

We can also load packages

```
using CairoMakie
using LaTeXStrings
```

and use them to make plots

```julia
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
```



4.2 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)

4.2.1 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"

4.2.2 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"

4.2.3 Word

You can also render the document to Microsoft Word. See the Quarto documentation.

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"