

Setting up Your Julia Environment

Jesse Perla, Thomas J. Sargent and John Stachurski

December 4, 2020

1 Contents

- Overview [2](#)
- A Note on Jupyter [3](#)
- Desktop Installation of Julia and Jupyter [4](#)
- Using Julia on the Web [5](#)
- Installing Packages [6](#)

2 Overview

In this lecture we will cover how to get up and running with Julia.

There are a few different options for using Julia, including a [local desktop installation](#) and [Jupyter hosted on the web](#).

If you have access to a web-based Jupyter and Julia setup, it is typically the most straightforward way to get started.

3 A Note on Jupyter

Like Python and R, and unlike products such as Matlab and Stata, there is a looser connection between Julia as a programming language and Julia as a specific development environment.

While you will eventually use other editors, there are some advantages to starting with the [Jupyter](#) environment while learning Julia.

- The ability to mix formatted text (including mathematical expressions) and code in a single document.
- Nicely formatted output including tables, figures, animation, video, etc.
- Conversion tools to generate PDF slides, static HTML, etc.
- [Online Jupyter](#) may be available, and requires no installation.

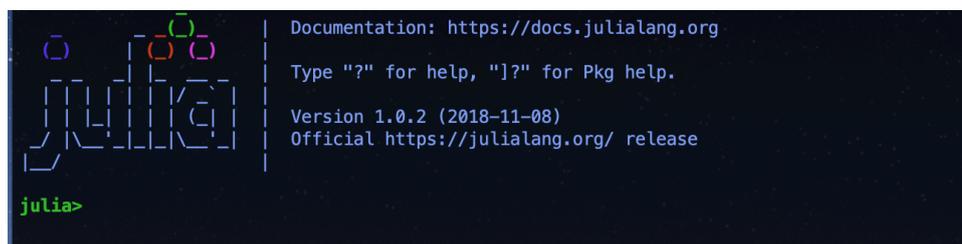
We'll discuss the workflow on these features in the [next lecture](#).

4 Desktop Installation of Julia and Jupyter

If you want to install these tools locally on your machine

- Download and install Julia, from [download page](#) , accepting all default options.
 - We do **not** recommend [JuliaPro](#).
- Open Julia, by either
 1. Navigating to Julia through your menus or desktop icons (Windows, Mac), or
 2. Opening a terminal and typing `julia` (Linux; to set this up on Mac, see end of section)

You should now be looking at something like this



This is called the JULIA *REPL* (Read-Evaluate-Print-Loop), which we discuss more [later](#).

- In the Julia REPL, hit `]` to enter package mode and then enter.

```
add IJulia InstantiateFromURL
```

This adds packages for

- The `IJulia` kernel which links Julia to Jupyter (i.e., allows your browser to run Julia code, manage Julia packages, etc.).
- The `InstantiateFromURL` which is a tool written by the QE team to manage package dependencies for the lectures.

Note: To set up the Julia terminal command on Mac, open a terminal and run `sudo ln -s <where_julia_app_is>/Contents/Resources/julia/bin/julia /usr/local/bin/julia`.

The full command might look like `sudo ln -s /Applications/Julia-1.4.app/Contents/Resources/julia/bin/julia /usr/local/bin/julia`, if you placed the app in your `Applications` folder.

Note: To obtain the full set of packages we use, at this stage you can run the following (see [the package setup section](#).)

```
using InstantiateFromURL
github_project("QuantEcon/quantecon-notebooks-julia", version = "0.8.0", instant
```

4.1 Installing Jupyter

If you have previously installed Jupyter (e.g., installing Anaconda Python by downloading the binary <https://www.anaconda.com/download/>) then the `add IJulia` installs everything you need into your existing environment.

Otherwise - or in addition - you can install it directly from the Julia REPL

```
using IJulia; jupyterlab()
```

Choose the default, `y` if asked to install Jupyter and then JupyterLab via Conda.

After the installation, a JupyterLab tab should open in your browser.

(Optional) To enable launching JupyterLab from a terminal, use [add Julia's Jupyter to your path](#).

4.2 Starting Jupyter

Next, let's install the QuantEcon lecture notes to our machine and run them (for more details on the tools we'll use, see our lecture on [version control](#)).

1. Install [git](#).
2. (Optional, but strongly recommended) Install the [GitHub Desktop](#).

4.2.1 GitHub Desktop Approach

After installing the Git Desktop application, click [this link](#) on your desktop computer to automatically install the notebooks.

It should open a window in the GitHub desktop app like this

Clone a Repository X

GitHub.com	GitHub Enterprise Server	URL
------------	--------------------------	-----

Repository URL or GitHub username and repository
(hubot/cool-repo)

Local Path

Choose a path you like and clone the repo.

Note: the workflow will be easiest if you clone the repo to the default location relative to the home folder for your user.

From a Julia REPL, start JupyterLab by executing

```
using IJulia; jupyterlab()
```

Alternatively, if you installed Jupyter separately in [Jupyter Installation](#) or [added Jupyter to your path](#) then run `jupyter lab` in your terminal.

Navigate to the location you stored the lecture notes, and open the [Interacting with Julia](#) notebook to explore this interface and start writing code.

4.2.2 Git Command Line Approach

If you do not wish to install the GitHub Desktop, you can get the notebooks using the Git command-line tool.

Open a new terminal session and run

```
git clone https://github.com/quantecon/quantecon-notebooks-julia
```

This will download the repository with the notebooks in the working directory.

Then, `cd` to that location in your Mac, Linux, or Windows PowerShell terminal

```
cd quantecon-notebooks-julia
```

Then, either using the `using IJulia; jupyterlab()` or execute `jupyter lab` within your shell.

And open the [Interacting With Julia](#) lecture (the file `julia_environment.ipynb` in the list of notebooks in JupyterLab) to continue.

5 Using Julia on the Web

If you have access to an online Julia installation, it is the easiest way to get started.

Eventually, you will want to do a [local installation](#) in order to use other [tools and editors](#) such as [Atom/Juno](#), but don't let the environment get in the way of learning the language.

5.1 Using Julia with JupyterHub

If you have access to a web-based solution for Jupyter, then that is typically a straightforward option

- Students: ask your department if these resources are available.
- Universities and workgroups: email [\[mailto:contact@quantecon.org\]](mailto:contact@quantecon.org){contact@quantecon.org} for help on setting up a shared JupyterHub instance with precompiled packages ready for these lecture notes.

5.1.1 Obtaining Notebooks

Your first step is to get a copy of the notebooks in your JupyterHub environment.

While you can individually download the notebooks from the website, the easiest way to access the notebooks is usually to clone the repository with Git into your JupyterHub environment.

JupyterHub installations have different methods for cloning repositories, with which you can use the url for the notebooks repository: <https://github.com/QuantEcon/quantecon-notebooks-julia>.

6 Installing Packages

After you have some of the notebooks available, as in [above](#), these lectures depend on functionality (like packages for plotting, benchmarking, and statistics) that are not installed with every Jupyter installation on the web.

If your online Jupyter does not come with QuantEcon packages pre-installed, you can install the `InstantiateFromURL` package, which is a tool written by the QE team to manage package dependencies for the lectures.

To add this package, in an online Jupyter notebook run (typically with `<Shift-Enter>`)

```
In [1]: ] add InstantiateFromURL
```

Then, run

```
using InstantiateFromURL
github_project("QuantEcon/quantecon-notebooks-julia", version = "0.8.0", instant
```

If your online Jupyter environment does not have the packages pre-installed, it may take 15-20 minutes for your first QuantEcon notebook to run.

After this step, open the downloaded [Interacting with Julia](#) notebook to begin writing code.

If the QuantEcon notebooks do not work after this installation step, you may need to speak to the JupyterHub administrator.