Getting set-up in JupyterLab

Oliver Will
2021-02-18

I typically use R through RStudio. One can probably use RStudio to run both Python and Julia now, but I feel there's an interest in this group in using Jupyter Notebooks. I'm a Windows user, so here's how I set up Jupyter Lab for Windows.

Install the Python and Julia REPL or console apps

I like to start with the REPL or console apps that come from the programming project themselves. I used to be able to access the R console app, but no longer can. I didn't anticipate that the console apps would be useful, which it was for Julia.

To install Python

Go to https://www.python.org/downloads/. The download button on the main webpage is for Python 3.9.1 for 64-bit Windows. I downloaded the installer and followed the instructions. Python has always been easy to install on Windows.

To install Julia

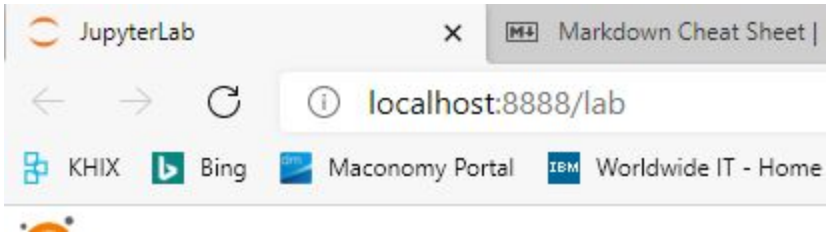
Go to https://julialang.org/downloads/. I chose the Windows 64-bit installer and followed the instructions.

Use Anaconda to manage JupyterLab

Jupyter Notebooks is written in Python. There are many ways to install Jupyter Notebooks with Python, and most require a Python package manager. I'm acquainted with the R package manager, but not Python's. I didn't want to manually install a Python manager, so I went with Anaconda Navigator. I tried Anaconda 4 years ago and had a rough time. Now it works much better. This website is a year old, but it has thorough instructions for installing Anaconda on Windows https://www.geeksforgeeks.org/how-to-install-anaconda-on-windows/ even though you can get more up-to-date versions of the underlying programs.

I went with Python 3.x for the Anaconda installation. My presentation is on toy problems and I no longer remember why one might want to go with 2.7. Also, Anaconda installed a second version of Python on my computer in my user directory.

One thing that made my set-up more straightforward is that I ran JupyterLab off my local computer. I can see this by looking at the URL line in my web browser.



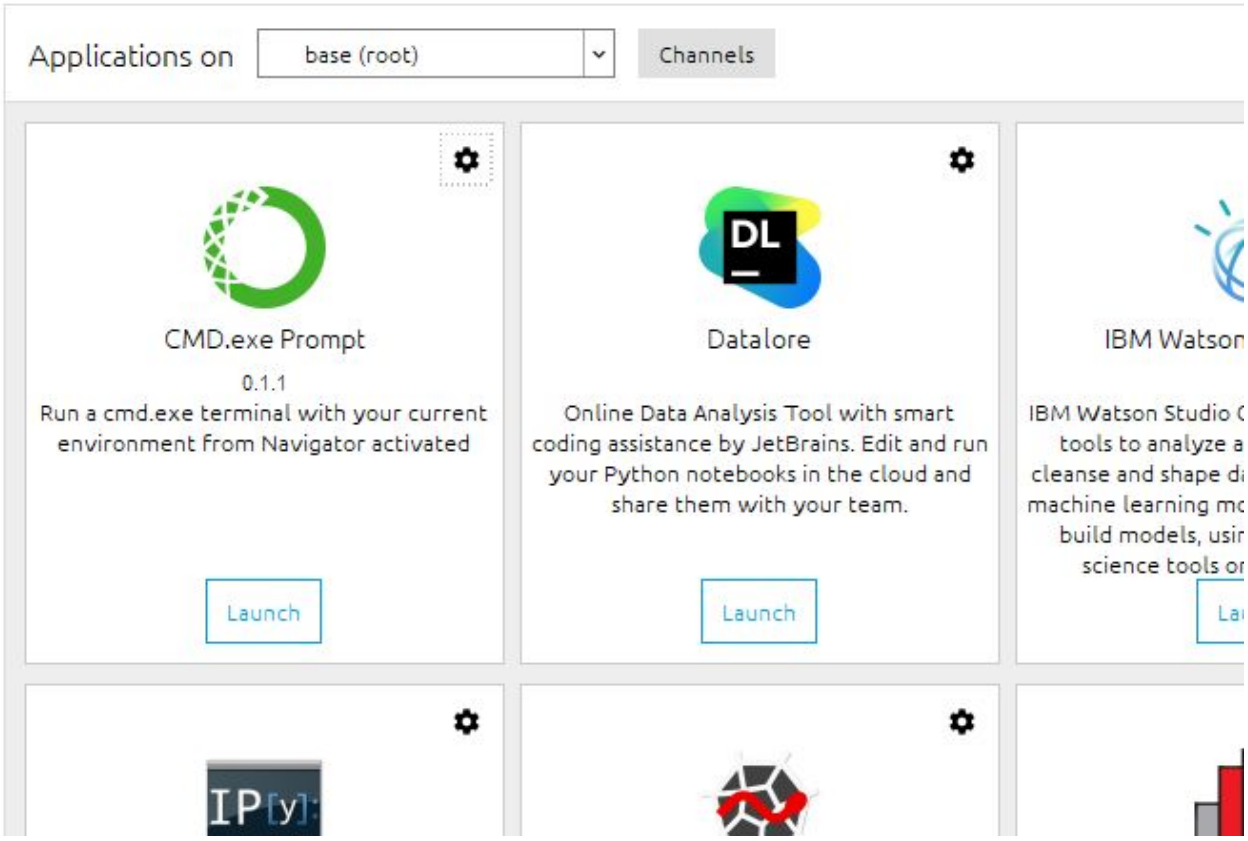
I only bring this up because once you are working on Hadoop or Spark, you shouldn't be on localhost. JupyterLab works well in Microsoft Edge.

Linking Jupyter Notebooks with Python, R, and Julia

My installation of Anaconda Navigator 1.10.0 has two base programs for running Jupyter Notebooks: Jupyter Notebooks and JupyterLab. I find JupyterLab more intuitive, so I went with it. Launching JupyterLab initially only has access to a Python kernel.

Adding the R kernel to JupyterLab

There seems to be a lot of ways to do this, but the easiest instructions were at https://datatofish.com/r-jupyter-notebook/. The only tricky part was that CMD.exe needs to be launched from within Anaconda Navigator and not the Windows start menu.



Working through the instructions takes a while, but is thoroughly documented. R version 3.6.5 is installed into the Anaconda system. My RStudio is using the latest version of R at 4.0.9. These are toy examples, so we'll be fine using an older version of R. A second version of R has been installed in my user profile.

Adding the Julia kernel to Jupyter

Again, Data to Fish had useful instructions https://datatofish.com/add-julia-to-jupyter/. The instructions are thorough and there's no need to repeat them here. One thing to note is that the installation needs to be done through the Julia console, so I'm glad I downloaded that first.

Bonus: Adding the SAS kernel to Jupyter

Project Jupyter, https://jupyter.org/, advertises that it can run over 40 types of kernels. Out of curiosity, I thought that I added a SAS kernel to my JupyterLab. I find it difficult to discuss SAS with general data scientists because not everyone has it, so I didn't intend to make it part of my presentation. I already have SAS installed because of my job. To install the kernel, go to the Anaconda command line. I used the following two commands:

conda install saspy

pip install sas_kernel

and you can use SAS in JupyterLab. After this, I get a SAS notebook and console icon that don't work, because I don't have the config file pointing at a batch executable version of SAS.

Additional Comments

I've been adding packages to R, Python, and Julia without keeping track of which ones. If there is a package you're missing, here are the general instructions to add it

For R

Issue the command from inside the R notebook

install.packages("package_name")

It looks like Anaconda will install from its internal servers.

For Python

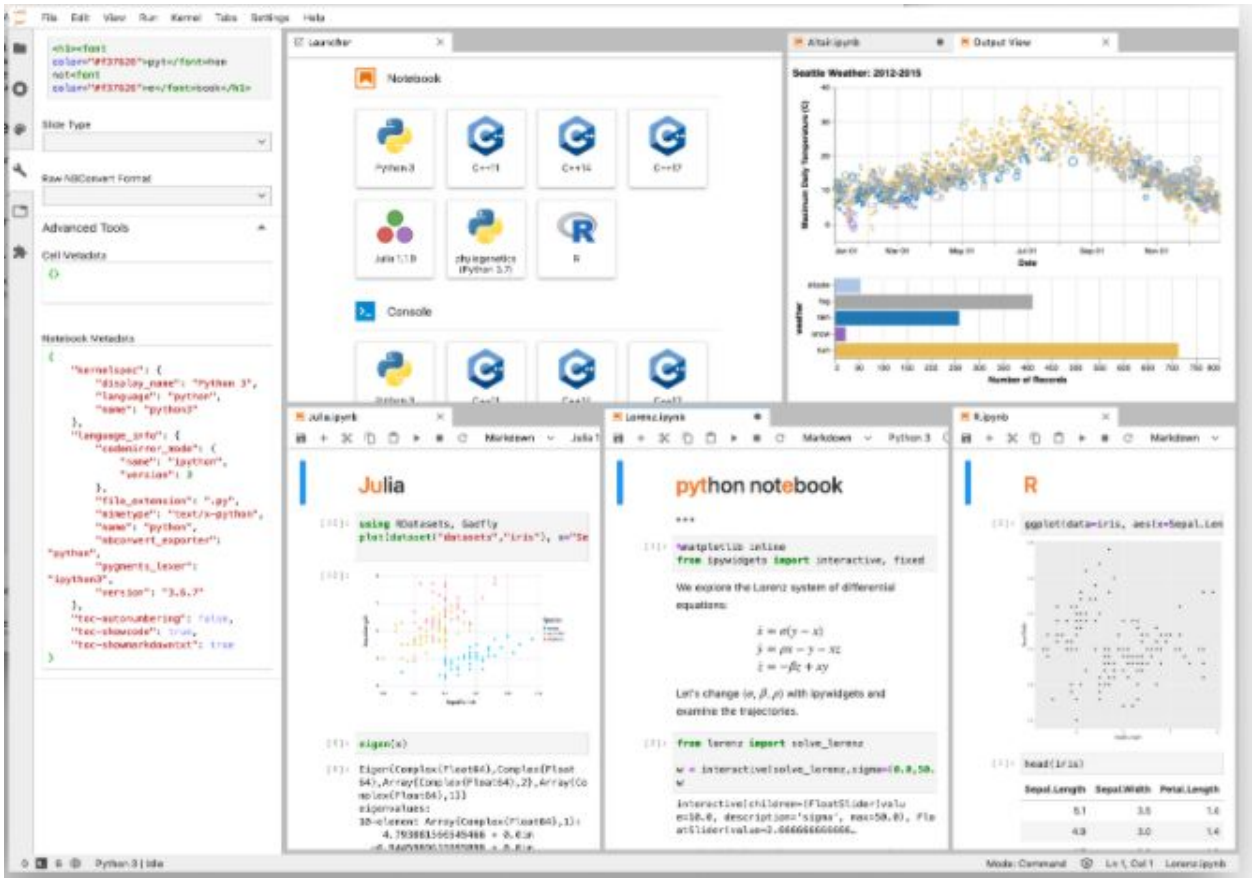
I don't think I've added a package beyond what came with the base installation. Conda and pip work from the Anaconda command line.

For Julia

Issue the following commands from within the Julia notebook

using Pkg
Pkg.add("package_name")

Finally, it looks like I might've subliminally taken the idea for this workshop from the Jupyter Project homepage. Here's a screenshot


Launcher
Notebook
Console
Advanced Tools
Cell Metadata
Notebook Metadata
Julia.ipynb
Lorenz.ipynb
R.ipynb
Output View
Seattle Weather: 2012-2015
Julia
python notebook
R
We explore the Lorenz system of differential equations:
Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.
Python 3 | Idle
Mode: Command