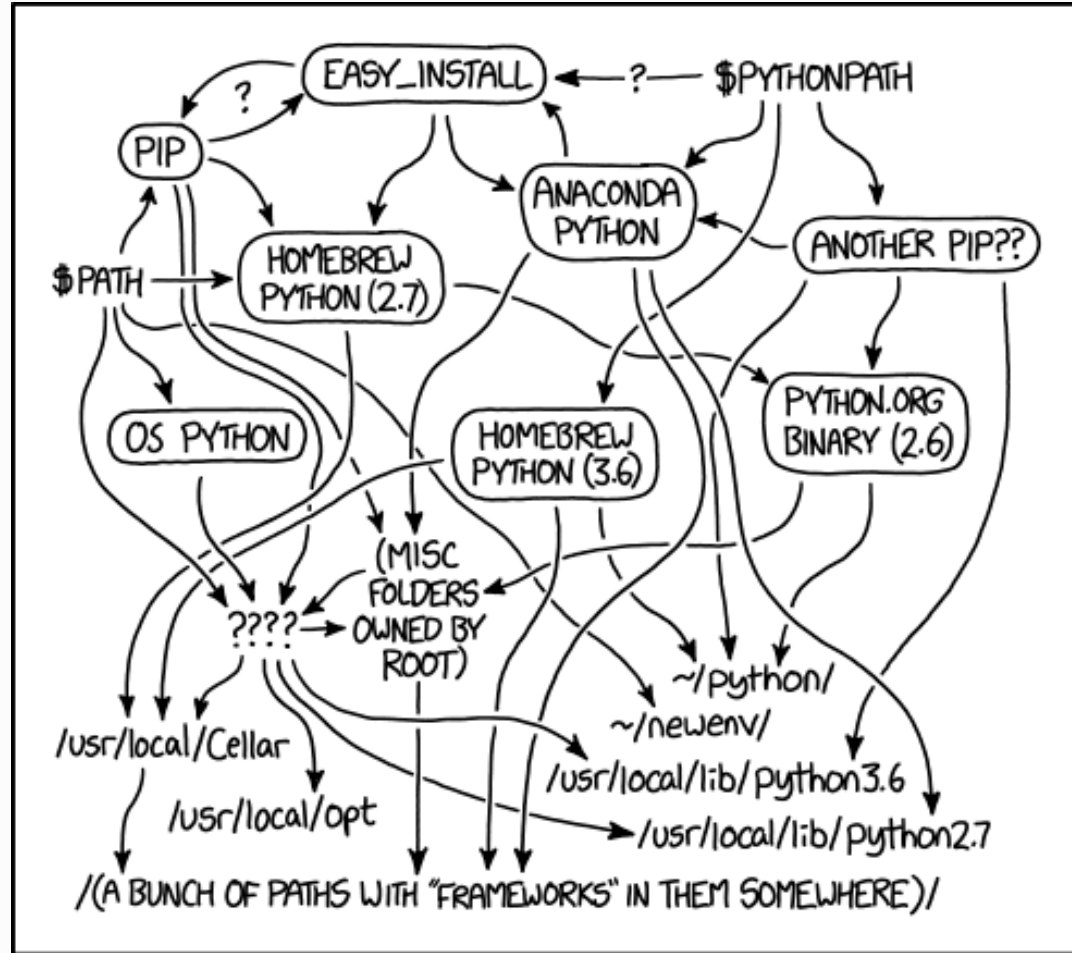




MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

CS 1671 / CS 2071

Human Language Technologies

Session 2: Python for data science

Michael Miller Yoder

August 27, 2026

About Hang Zhang (TA)

- **3rd year ISP PhD student**
- **LinkedIn:** <https://www.linkedin.com/in/hang-zhang-99ab2a16b/>
- **Research interests:**
 - AI Agent
 - AI in Healthcare
- **Office Hours:**
 - Tuesday 9-10am
 - Zoom link: <https://pitt.zoom.us/j/98748568131>
 - Email: haz269@pitt.edu



Course logistics

- Reading for next class
 - Jurafsky + Martin 2-2.6, 2.8, 2.10
- Please remind me of your name before asking or answering a question

Learning objectives: Python for data science

Students will be able to:

- Load data into a Pandas DataFrame
- Describe different data types in Pandas
- Select and subset data from Pandas DataFrames
- Write Python functions
- Apply vectorized functions to Pandas DataFrame columns
- Create Python projects with clearly defined environments using Pixi

JupyterHub setup

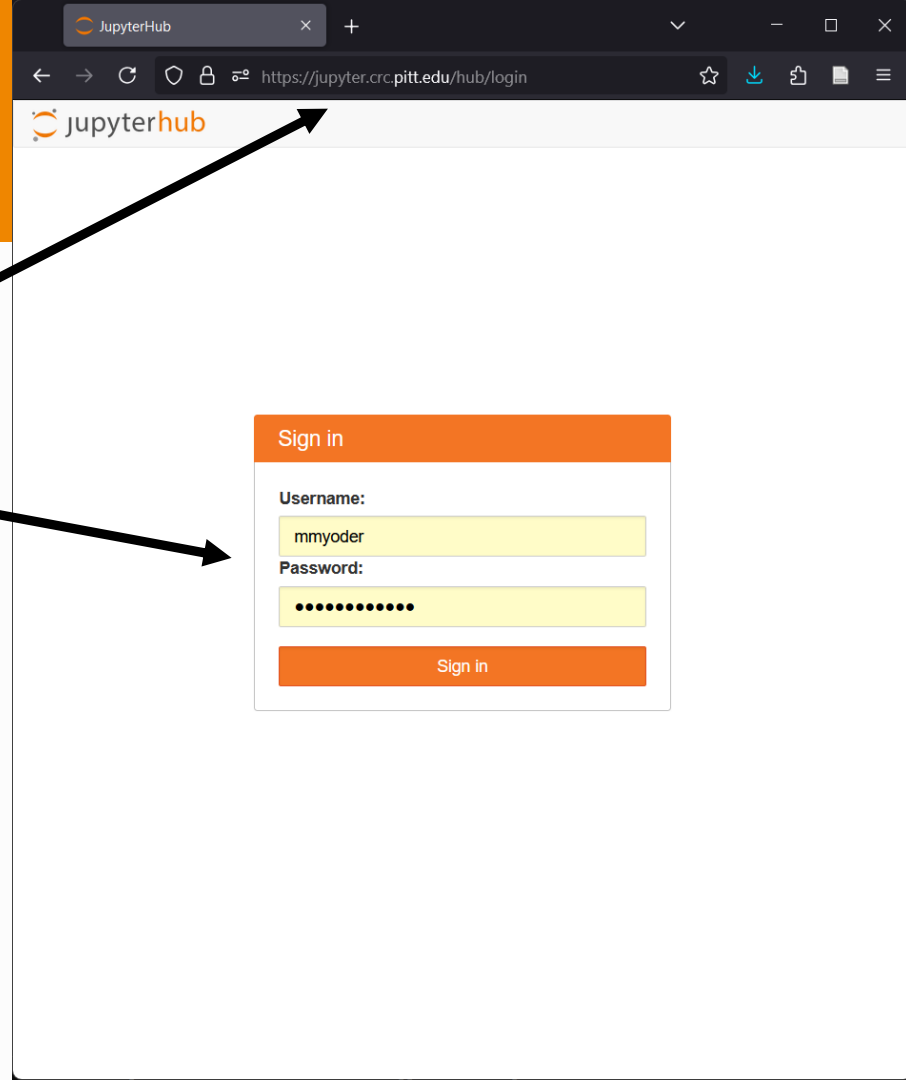
CRCD and JupyterHub

- CRCD (Center for Research Computing and Data) is a Pitt center providing computing services on various clusters
- They maintain a JupyterHub where people can run Jupyter Notebooks, which are interactive Python documents, on their servers
- What we will be using the CRCD for:
 - Working through code examples in class
 - Writing code to submit as part of homework assignments
 - Running code and storing data for your projects (if you want to)

Logging in to your CRCD JupyterHub account

1. Go to `jupyter.crc.pitt.edu` in a web browser
2. Log in with your Pitt credentials (lowercase)

Note that if you are off-campus, you have to log in to the Pitt VPN first through the GlobalProtect app. Instructions: <https://services.pitt.edu/TDClient/33/Portal/KB/ArticleDet?ID=3426>



Starting a Jupyter Notebook on the CRCD JupyterHub

1. Partition: TEACH – 6 CPUs – 45 GB
2. Under Select Virtual Environment, select Provide custom path
3. Custom Environment Path: `/ix1/cs1671-2026f/class-env/.pixi/envs/default`
4. Click Start
5. Wait for the server to start up

Server Options

The screenshot shows the 'JupyterHub Session Configuration' window. It contains several sections: 'Select Partition:' with a dropdown menu showing 'TEACH - 6 CPUs - 45GB'; 'Select Virtual Environment:' with a dropdown menu showing 'Provide custom path'; 'Custom Environment Path:' with a text input field containing the path `/ix1/cs1671-2026f/class-env/.pixi/envs/default`; 'Select Modules to Load:' with a list of modules (Amber 2024, Cuda 12.3, OpenJDK 21.0.2) and a note 'Hold Ctrl/Cmd to select multiple modules'; and an 'Account:' field with the text 'your class account'. At the bottom is a large orange 'Start' button. White arrows point from the list on the left to the 'TEACH' dropdown, the 'Provide custom path' dropdown, the custom environment path text field, and the 'Start' button.

JupyterHub Session Configuration

Select Partition:
TEACH - 6 CPUs - 45GB

Select Virtual Environment:
Provide custom path

Custom Environment Path:
`/ix1/cs1671-2026f/class-env/.pixi/envs/default`

Select Modules to Load:
Amber 2024
Cuda 12.3
OpenJDK 21.0.2
Hold Ctrl/Cmd to select multiple modules

Account: your class account

Start

Cancel the “Build Recommended” window

Build Recommended

JupyterLab build is suggested:

```
@jupyterlab/audio-extension needs to be removed from build
@jupyterlab/video-extension needs to be removed from build
@jupyterlab/application changed from ~4.6.3 to ~4.4.4
@jupyterlab/application-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/apputils-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/cell-toolbar-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/celltags-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/codemirror-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/completer-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/console-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/coreutils changed from ~6.6.3 to ~6.4.4
@jupyterlab/csvviewer-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/debugger-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/docmanager-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/documentsearch-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/extensionmanager-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/filebrowser-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/fileeditor-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/help-extension changed from ~4.6.3 to ~4.4.4
@jupyterlab/htmlviewer-extension changed from ~4.6.3 to ~4.4.4
```

Cancel

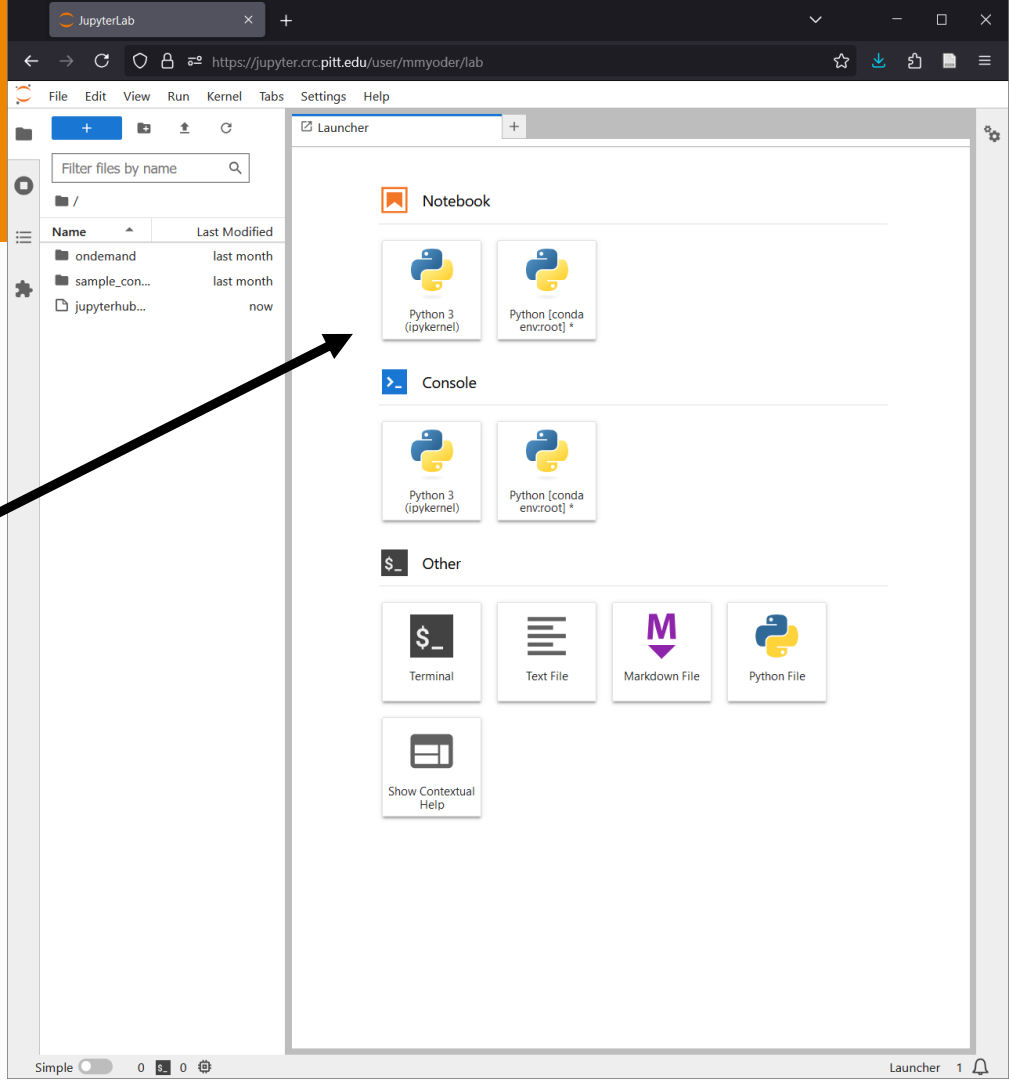
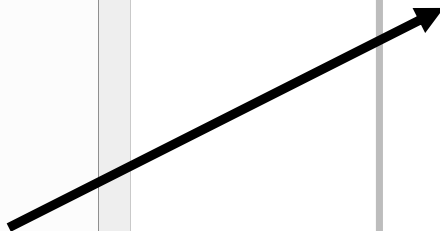
Build

Welcome to your JupyterLab

Files are here



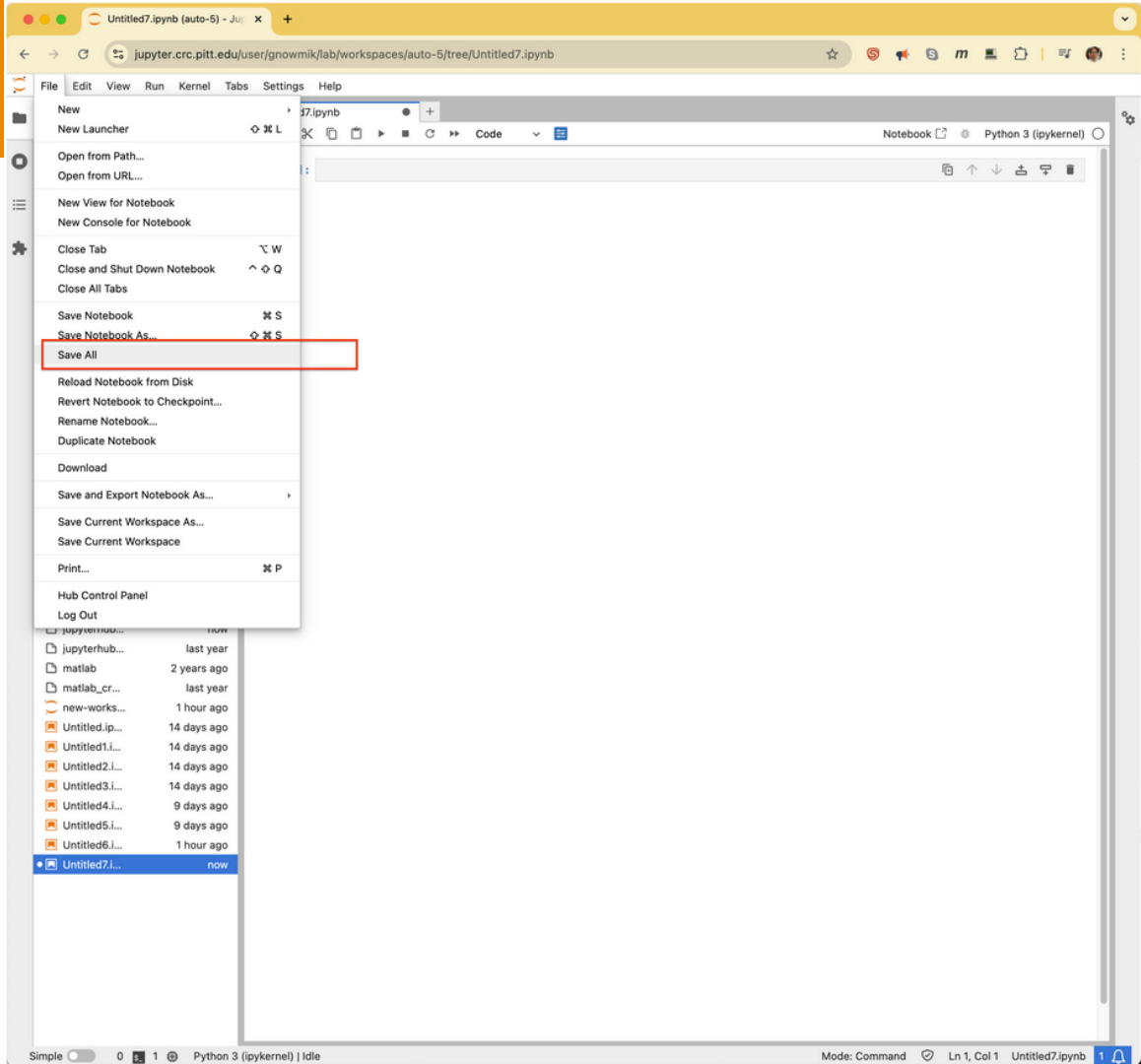
You can launch a new Jupyter Notebook by clicking Python 3 (ipykernel) under Notebook



Jupyter Notebook basics

- Each block is called a “cell”
 - Has input and possibly output
 - Input can be Python code, Markdown or shell commands (after !)
- Modes
 - Command mode
 - Move, select, manipulate cells
 - Get into command mode by clicking anywhere outside of a cell
 - Edit mode
 - Blinky cursor within a cell, which is highlighted with a blue border
 - Edit content of a particular cell
- Running cells
 - Click “Run” button or do Ctrl+Enter (on Windows or Linux, Cmd+Enter on Mac) to run code or render Markdown
 - Any result will be shown in the output of the cell

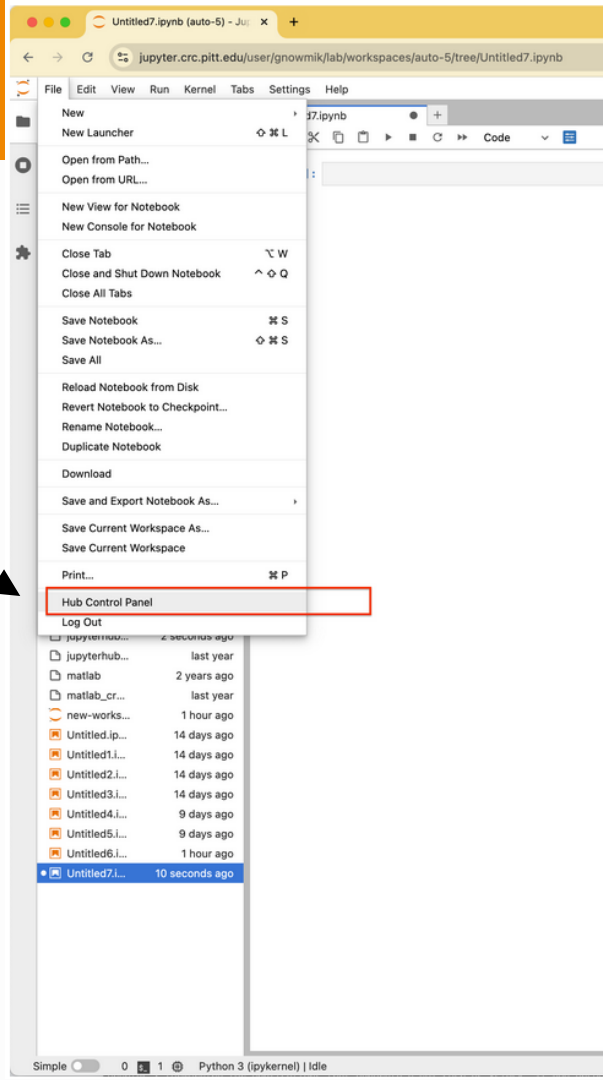
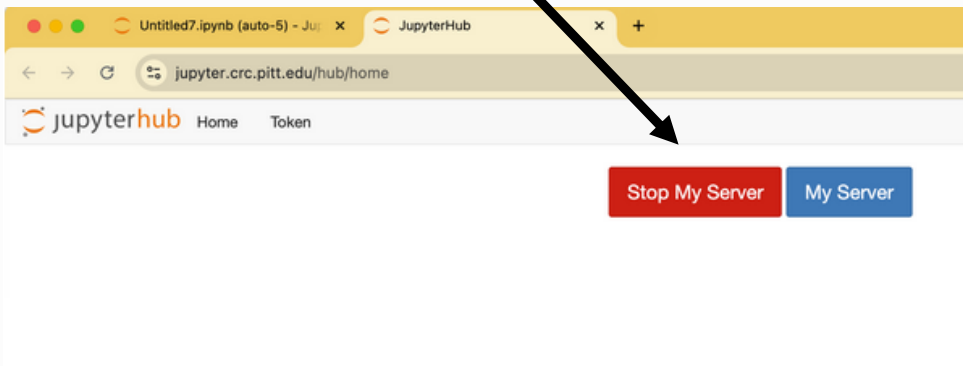
Saving your work



Ending your session

Be sure to save your work before ending the session

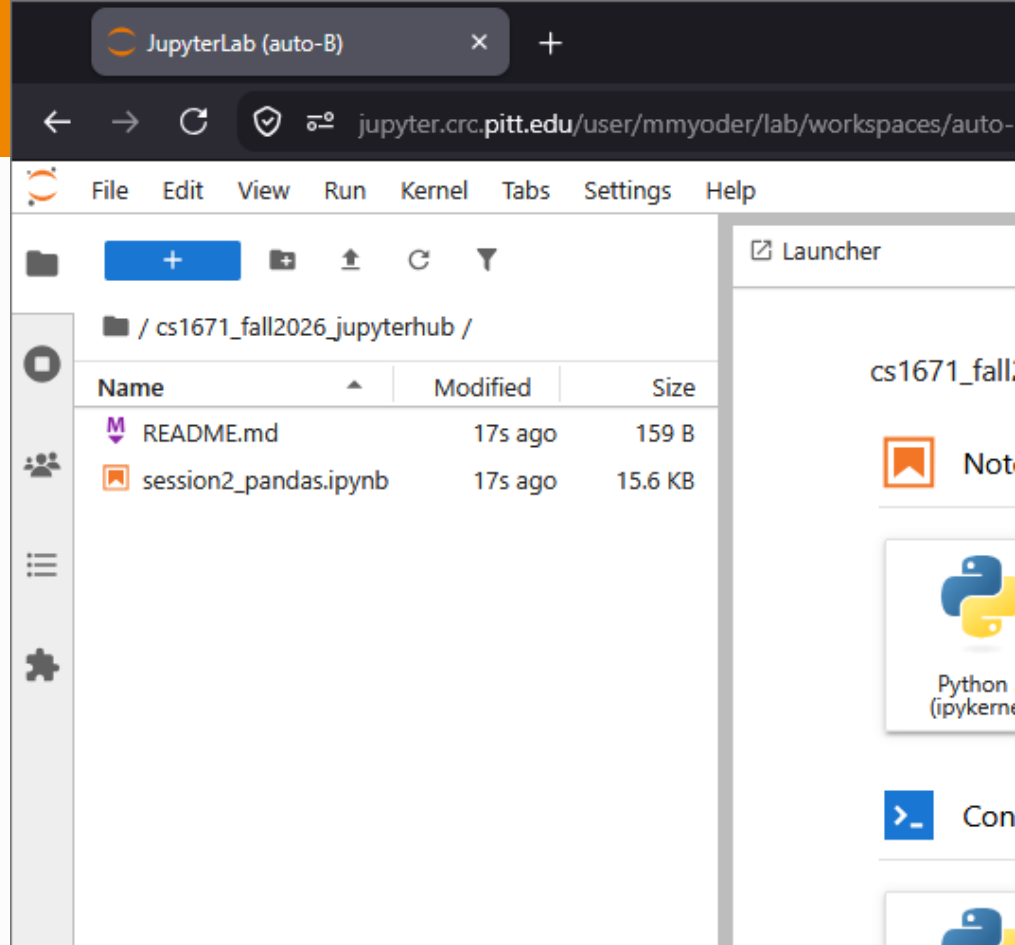
1. Select **File > Hub Control Panel**
2. Click **Stop My Server**



Coding activity: Pandas and Python basics

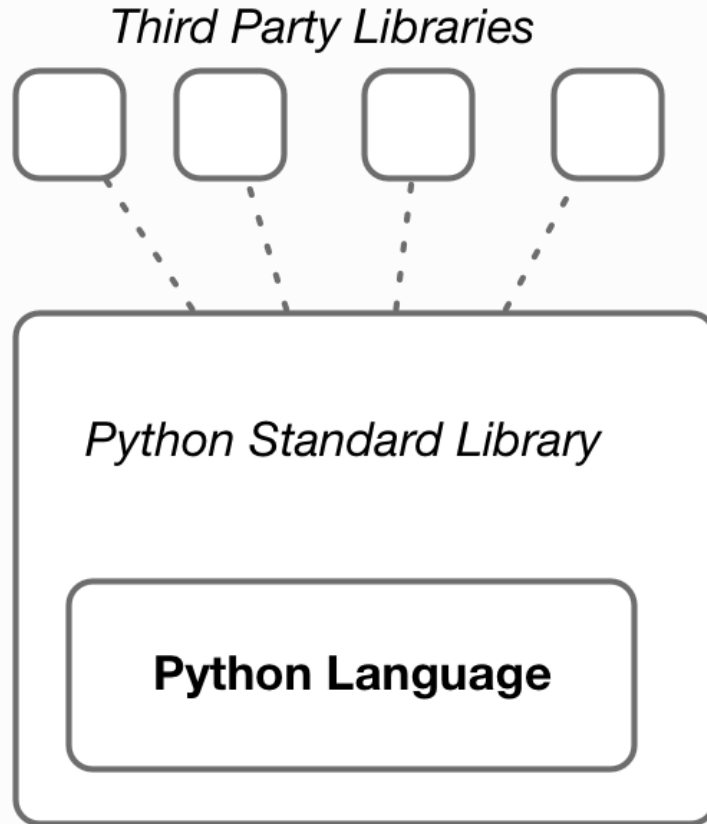
Load in-class notebooks

1. Go to this [nbgitpuller link](#) (also available on course website). This pulls in files from a GitHub repo into your JupyterHub
2. Open `session2_pandas.ipynb`



Keeping track of Python packages with projects and environments

Python packages



Python environments

- List of Python packages and versions of those packages used for a project
- Important for reproducibility
- It's best to isolate a unique environment for each project

Python package managers

- Tools for managing packages:
 - Downloading and installing packages
 - Checking package dependencies and any dependency conflicts
- Create and manage Python environments
- Default: pip to install packages, virtualenv (venv) for environments
- Other package managers with more functionality: uv, Anaconda, pixi, poetry, pyenv

Pixi package manager

- The class environment uses **pixi**
 - Fast
 - Creates a standard .toml file for reproducibility
 - Compatible with Anaconda (which the CRCD uses)



Install pixi

- macOS or Linux:
 - `curl -fsSL https://pixi.sh/install.sh | sh`
 - Or for macOS: `brew install pixi`
- Windows
 - Download installer:
<https://pixi.prefix.dev/latest/installation/>
- Restart your terminal

Creating new Python projects with pixi

- `pixi init project --format pyproject`
- `pixi add <package>`