

EEPS 440/640

Geospatial Data Science (GDS)

The primary class website for this course is on GitHub

https://github.com/NoemiVergopolan/EEPS_GeospatialData.git

The screenshot shows the GitHub interface for the repository **EEPS_GeospatialData** by user **NoemiVergopolan**. The repository is marked as **Private**. The top navigation bar includes links for **Code**, **Issues**, **Pull requests**, **Actions**, **Projects**, **Security**, **Insights**, and **Settings**. Below the repository name, there are buttons for **Unwatch** (1), **Fork** (0), and **Star** (0). The main content area shows the **main** branch with 1 branch and 0 tags. A search bar labeled "Go to file" is present. Below this, a list of files and folders is shown, all created by "setup main page":

File/Folder	Author	Time
Assignments	setup main page	10 minutes ago
Lectures	setup main page	now
Syllabus	setup main page	10 minutes ago
README.md	setup main page	5 minutes ago

Below the file list, the **README** file is selected, showing the title **EEPS 440/640 - Geospatial Data Science**. On the right sidebar, the **About** section states "EEPS 440/460 Geospatial Data Science" and lists 0 stars, 1 watching, and 0 forks. The **Releases** section indicates "No releases published" with a link to "Create a new release". The **Packages** section indicates "No packages published".

Now you have your own repository in GitHub that you will use to submit your HWs

https://github.com/UserName/GDS25_Initials

The screenshot shows a GitHub repository page for 'GDS25_nvr' by user 'NoemiVergopolan'. The repository is private. The main content area shows a commit by 'Noemi Vergopolan Da Rocha' adding a file 'HW0.ipynb'. Below the commit, there is a 'README' section with a prompt to 'Add a README' and a green button labeled 'Add a README'. The right sidebar contains metadata: 'About' (no description), 'Activity' (0 stars, 1 watching, 0 forks), 'Releases' (no releases published), 'Packages' (no packages published), and 'Languages' (Jupyter Notebook 100.0%).

Navigation bar: NoemiVergopolan / GDS25_nvr

Repository: GDS25_nvr (Private)

Commit: Noemi Vergopolan Da Rocha add HW0 file a2e2e4a · 43 minutes ago

File: HW0.ipynb add HW0 file 43 minutes ago

README: Add a README

Activity: 0 stars, 1 watching, 0 forks

Releases: No releases published. [Create a new release](#)

Packages: No packages published. [Publish your first package](#)

Languages: Jupyter Notebook 100.0%

Lets create a token password for YOUR repository

Log to your Github Account

<https://github.com/settings/personal-access-tokens>

1. Generate new token
2. Give it any name (e.g., GDS)
3. Set it to unlimited expiration
4. Only selected repositories: GDS25_Initials
5. Repository Permissions → Read and write for all
6. Account Permissions → Read and write for all
7. Finish with: Generate Token

Save your token password somewhere safe

How to access the HPC

Internet access on campus:

- Make sure you are connected to **Rice Owls** wifi

Internet access off campus:

- You need Rice VPN. Follow the steps here:
<https://kb.rice.edu/page.php?id=82263>
- If you have issues, please email:
helpdesk@rice.edu

How to access the HPC

- Check the HPC account info you received
- Open the terminal on your machine

This week → `ssh netID@notsx.rice.edu`

Next week onwards → `ssh netID@nots.rice.edu`

- Password and login you use for Rice

Launch a job on the HPC

Copy the job submission script template to your home
(you only need to do this once)

```
cp /projects/eeps440/jupyter-smp.slurm /home/netID
```

Go to your home folder and launch your job

```
cd /home/netID  
sbatch jupyter-smp.slurm
```

```
[nv25@loginx1 ~]$ sbatch jupyter-smp.slurm  
Submitted batch job 229150
```

- See your job in the queue with **squeue -u netID**

```
[nv25@loginx1 ~]$ squeue -u nv25
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
229150	commons	jupyter-	nv25	R	1:12	1	bb5u26c1

- Use **ls** to look for the job output **slurm-JOBID.out**

```
[nv25@loginx1 ~]$ ls
~  jupyter-smp.slurm  slurm-229150.out
```

Pro-tip: **ls -lrt** will list the files in the folder sorted by time

- Use **cat** **slurm-JOBID.out** to show what is inside the job script output

```
[nv25@loginx1 ~]$ cat slurm-229150.out
```



```
[nv25@loginx1 ~]$ cat slurm-229150.out
```

Run the following command to set up an ssh tunnel to the compute node:

```
ssh -NL 60560:bb5u26c1:60560 nv25@nots.crc.rice.edu
```

```
ssh -NL 60560:bb5u26c1:60560 nv25@nots.crc.rice.edu
```

```
[I 2025-01-23 11:23:59.287 ServerApp] jupyterlab | extension was successfully linked.
[W 2025-01-23 11:23:59.287 NotebookApp] 'ip' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2025-01-23 11:23:59.287 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2025-01-23 11:23:59.287 NotebookApp] 'port' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2025-01-23 11:23:59.287 NotebookApp] 'allow_origin' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2025-01-23 11:23:59.288 ServerApp] nbclassic | extension was successfully linked.
[I 2025-01-23 11:23:59.612 ServerApp] notebook_shim | extension was successfully linked.
[I 2025-01-23 11:23:59.612 ServerApp] panel.io.jupyter_server_extension | extension was successfully linked.
[I 2025-01-23 11:23:59.700 ServerApp] notebook_shim | extension was successfully loaded.
[I 2025-01-23 11:23:59.701 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2025-01-23 11:23:59.702 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2025-01-23 11:23:59.705 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.12/site-packages/jupyterlab
[I 2025-01-23 11:23:59.705 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 2025-01-23 11:23:59.705 LabApp] Extension Manager is 'pypi'.
[I 2025-01-23 11:23:59.806 ServerApp] jupyterlab | extension was successfully loaded.
[I 2025-01-23 11:23:59.811 ServerApp] nbclassic | extension was successfully loaded.
[I 2025-01-23 11:23:59.812 ServerApp] panel.io.jupyter_server_extension | extension was successfully loaded.
[I 2025-01-23 11:23:59.812 ServerApp] Serving notebooks from local directory: /home/nv25
[I 2025-01-23 11:23:59.812 ServerApp] Jupyter Server 2.15.0 is running at:
[I 2025-01-23 11:23:59.812 ServerApp] http://bb5u26c1:60560/lab?token=f54ed35c711f6ffee18706435ab6913499c2b86c2fc45d47
[I 2025-01-23 11:23:59.812 ServerApp] http://127.0.0.1:60560/lab?token=f54ed35c711f6ffee18706435ab6913499c2b86c2fc45d47
[I 2025-01-23 11:23:59.812 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2025-01-23 11:23:59.816 ServerApp]
```

To access the server, open this file in a browser:

file:///home/nv25/.local/share/jupyter/runtime/jpserver-3571747-open.html

Or copy and paste one of these URLs:

```
http://bb5u26c1:60560/lab?token=f54ed35c711f6ffee18706435ab6913499c2b86c2fc45d47
http://127.0.0.1:60560/lab?token=f54ed35c711f6ffee18706435ab6913499c2b86c2fc45d47
```

```
http://127.0.0.1:60560/lab?token=f54ed35c711f6ffee18706435ab6913499c2b86c2fc45d47
```

```
[nv25@loginx1 ~]$
```

Open a **NEW terminal** and paste the tunnel command

```
ssh -NL 60560:bb5u26c1:60560 nv25@notes.crc.rice.edu
```

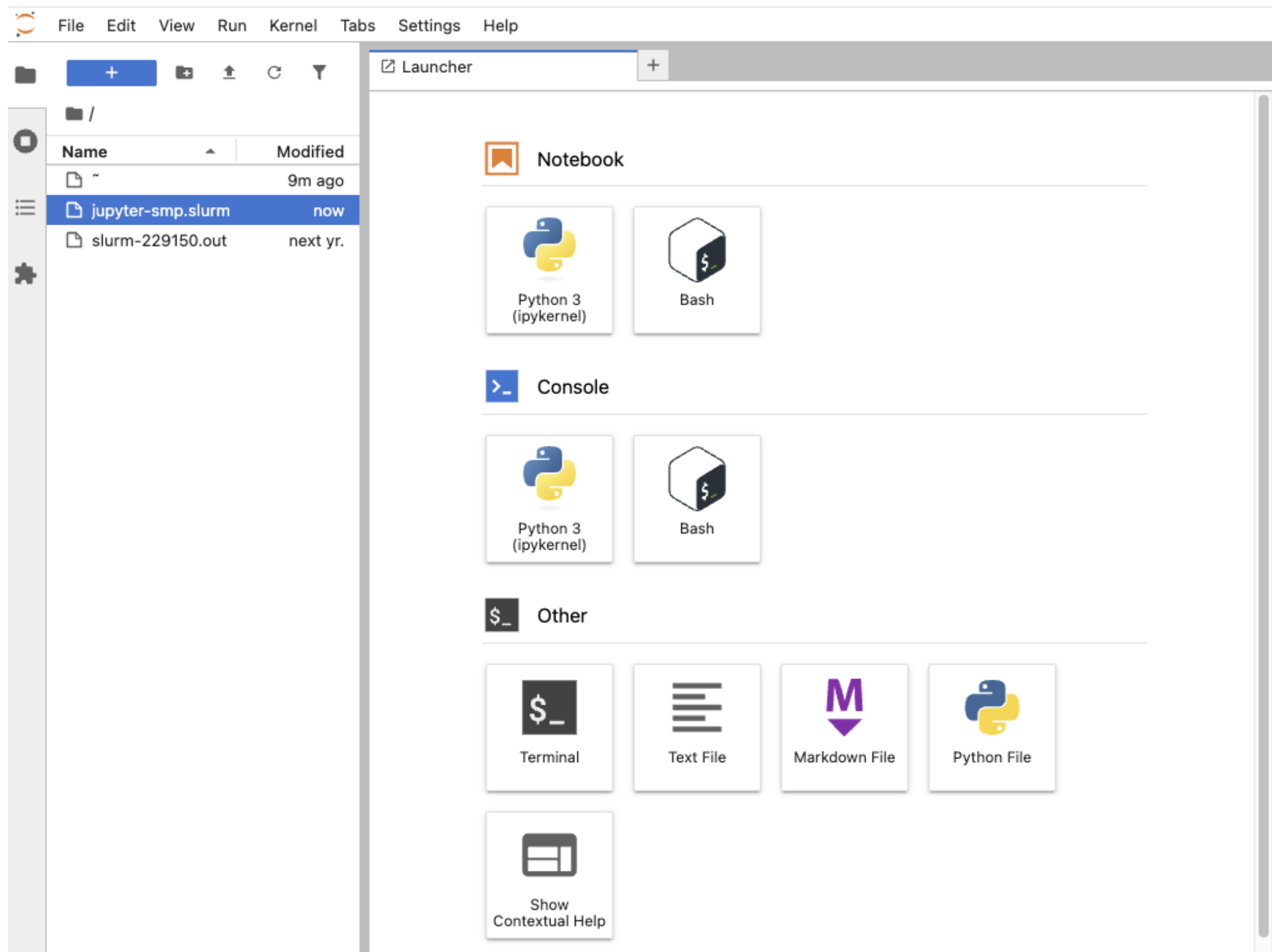
Type the password and let it be

Open your **internet browser** and paste the link starting with 127.0.....

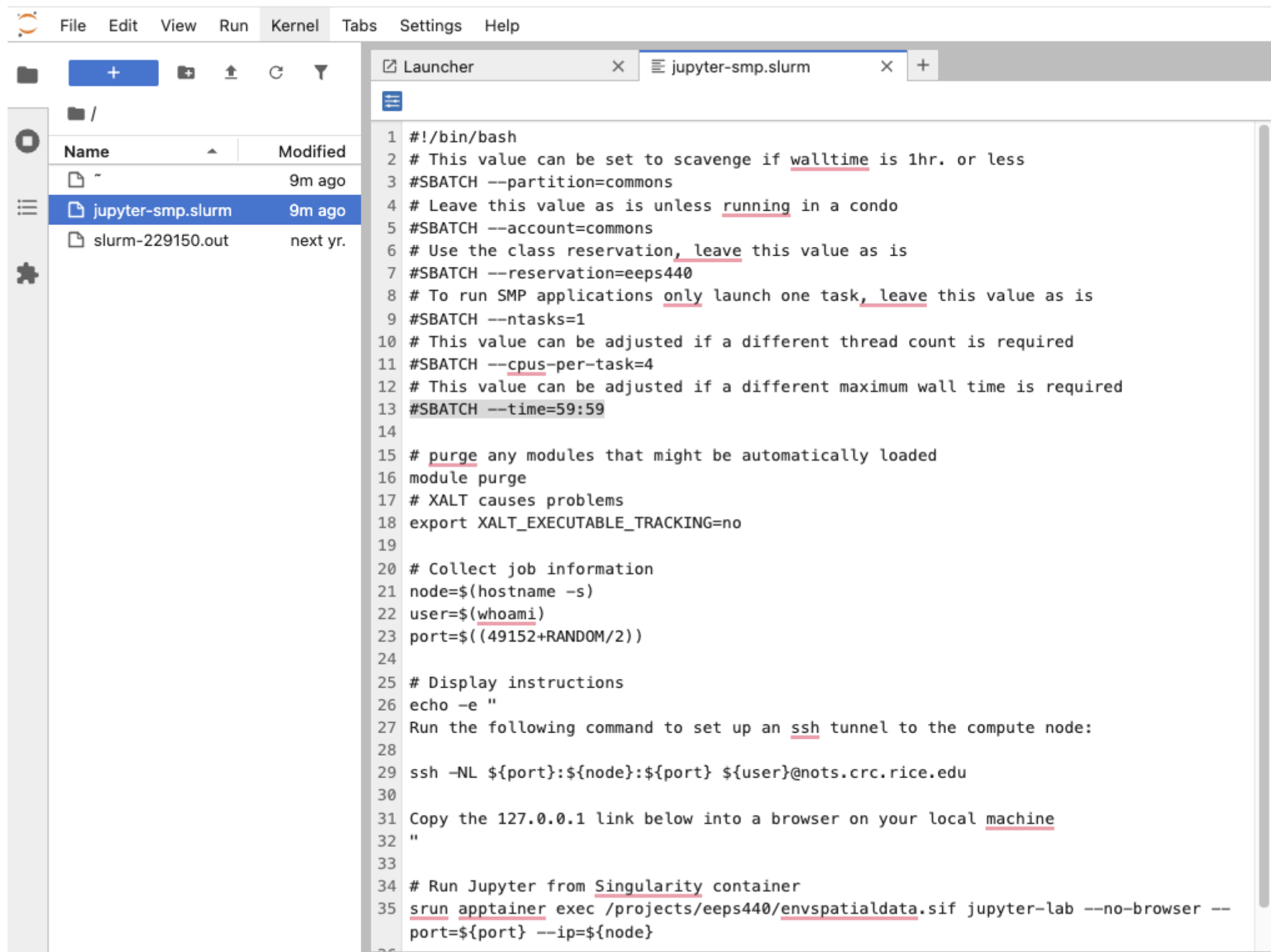
```
http://127.0.0.1:60560/lab?token=f54ed35c711f6ffee18706435ab6913499c2b86c2fc45d47
```

You are running your Jupyter lab on Rice's HPC!

Your browser should look like this:



Your job submission script template (jupyter-smp.slurm) defaults to 30min run. Let's change it to 59:59 min...



The screenshot shows the JupyterLab interface with the file explorer on the left and the code editor on the right. The file explorer shows a directory with files: `~` (9m ago), `jupyter-smp.slurm` (9m ago), and `slurm-229150.out` (next yr.). The code editor shows the content of `jupyter-smp.slurm`, which is a SLURM job submission script template. The script includes comments and SBATCH directives for partition, account, reservation, tasks, cpus, and time. The time directive is highlighted in the image.

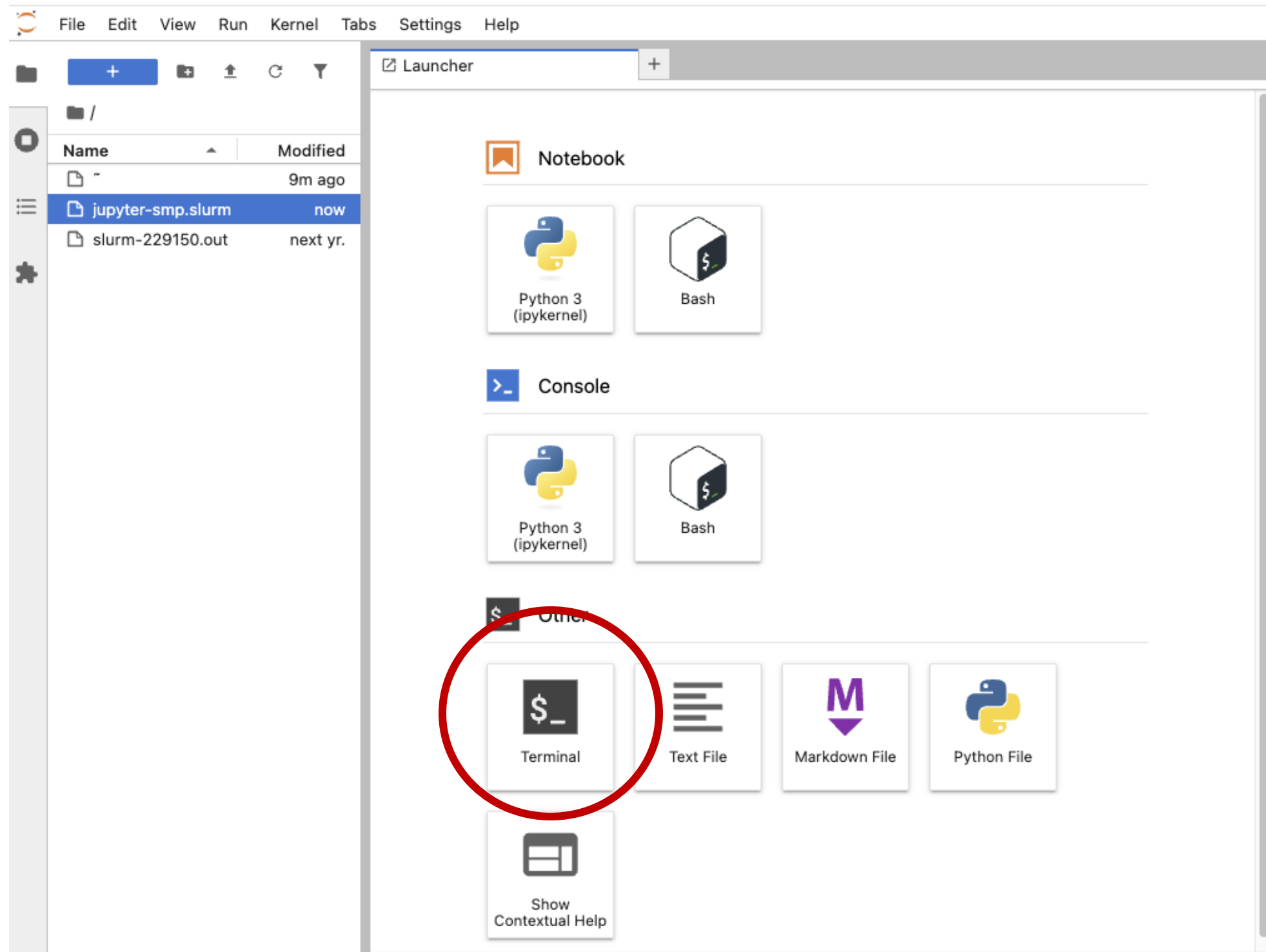
```
1 #!/bin/bash
2 # This value can be set to scavenge if walltime is 1hr. or less
3 #SBATCH --partition=commons
4 # Leave this value as is unless running in a condo
5 #SBATCH --account=commons
6 # Use the class reservation, leave this value as is
7 #SBATCH --reservation=eeps440
8 # To run SMP applications only launch one task, leave this value as is
9 #SBATCH --ntasks=1
10 # This value can be adjusted if a different thread count is required
11 #SBATCH --cpus-per-task=4
12 # This value can be adjusted if a different maximum wall time is required
13 #SBATCH --time=59:59
14
15 # purge any modules that might be automatically loaded
16 module purge
17 # XALT causes problems
18 export XALT_EXECUTABLE_TRACKING=no
19
20 # Collect job information
21 node=$(hostname -s)
22 user=$(whoami)
23 port=$((49152+RANDOM/2))
24
25 # Display instructions
26 echo -e "
27 Run the following command to set up an ssh tunnel to the compute node:
28
29 ssh -NL ${port}:${node}:${port} ${user}@nots.crc.rice.edu
30
31 Copy the 127.0.0.1 link below into a browser on your local machine
32 "
33
34 # Run Jupyter from Singularity container
35 srun apptainer exec /projects/eeps440/envspatialdata.sif jupyter-lab --no-browser --
36 port=${port} --ip=${node}
```

You can ask for more run time,
but you might sit longer in the
queue...

You can learn more about SLURM basics settings here:
<https://blog.ronin.cloud/slurm-intro/>

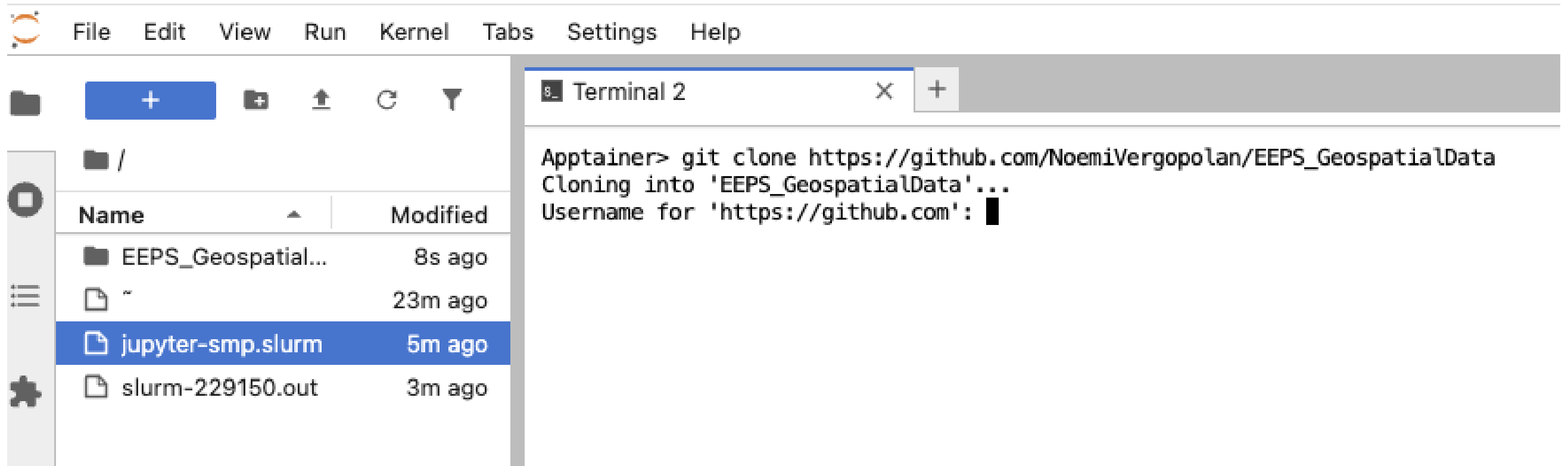
Let's get started:

Open a terminal on Jupyter Lab



git clone

https://github.com/NoemiVergopolan/EEPS_GeospatialData.git



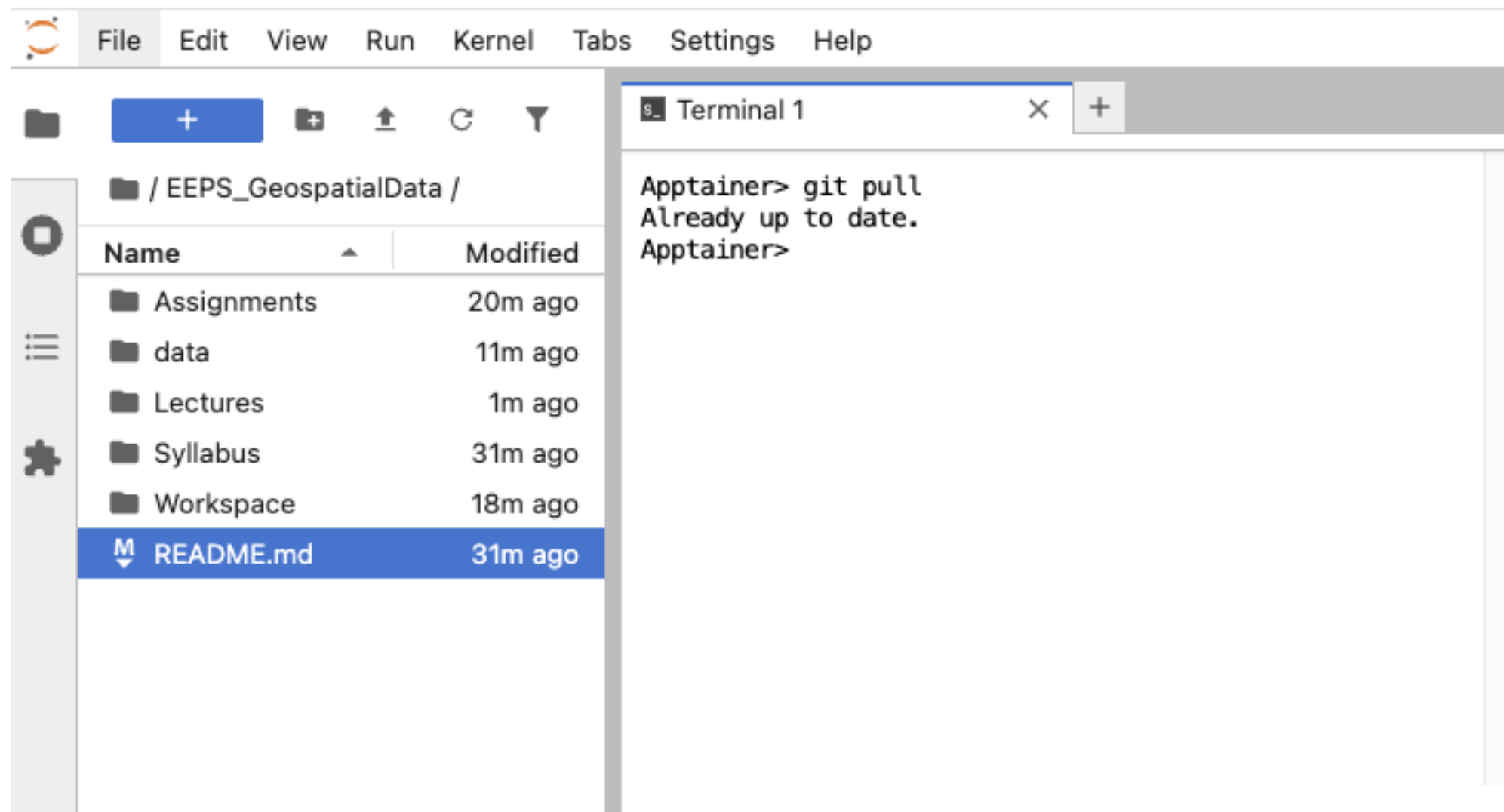
The screenshot shows a JupyterLab environment. On the left is a file browser with a sidebar containing icons for home, recent, and search. The main area shows a file list with columns 'Name' and 'Modified'.

Name	Modified
EEPS_Geospatial...	8s ago
~	23m ago
jupyter-smp.slurm	5m ago
slurm-229150.out	3m ago

On the right is a terminal window titled 'Terminal 2'. It contains the following text:

```
Apptainer> git clone https://github.com/NoemiVergopolan/EEPS_GeospatialData
Cloning into 'EEPS_GeospatialData'...
Username for 'https://github.com':
```

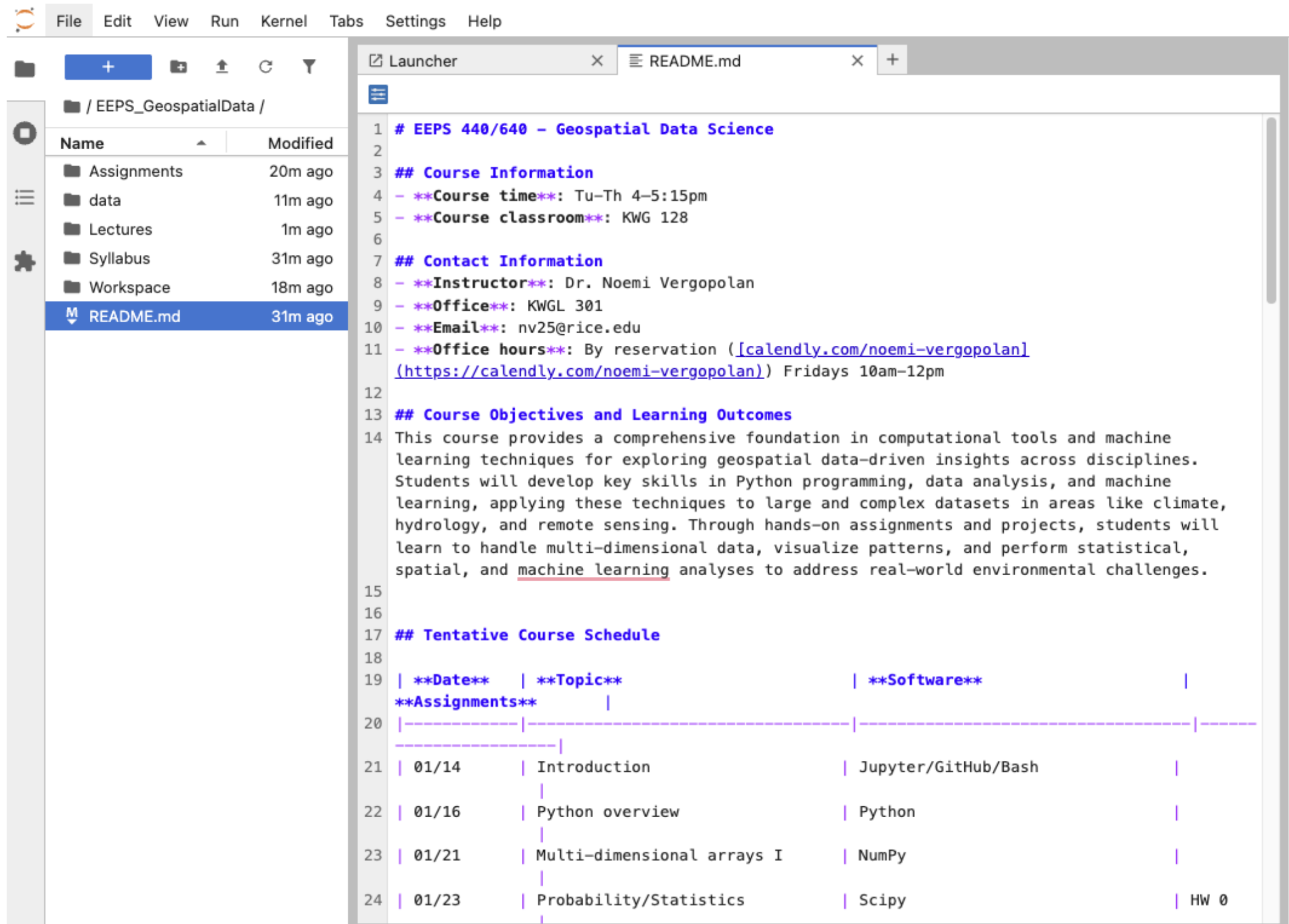
Git pull



Press enter after writing “git pull”

This is how you will have update the class repository on your system (i.e., as lectures and assignments are added online)

Explore the cloned directory



The screenshot displays a JupyterLab environment. On the left, a file explorer shows the directory structure of a cloned repository named 'EEPS_GeospatialData'. The files listed are 'Assignments' (modified 20m ago), 'data' (11m ago), 'Lectures' (1m ago), 'Syllabus' (31m ago), 'Workspace' (18m ago), and 'README.md' (31m ago). The 'README.md' file is selected and its content is displayed in the main editor area on the right.

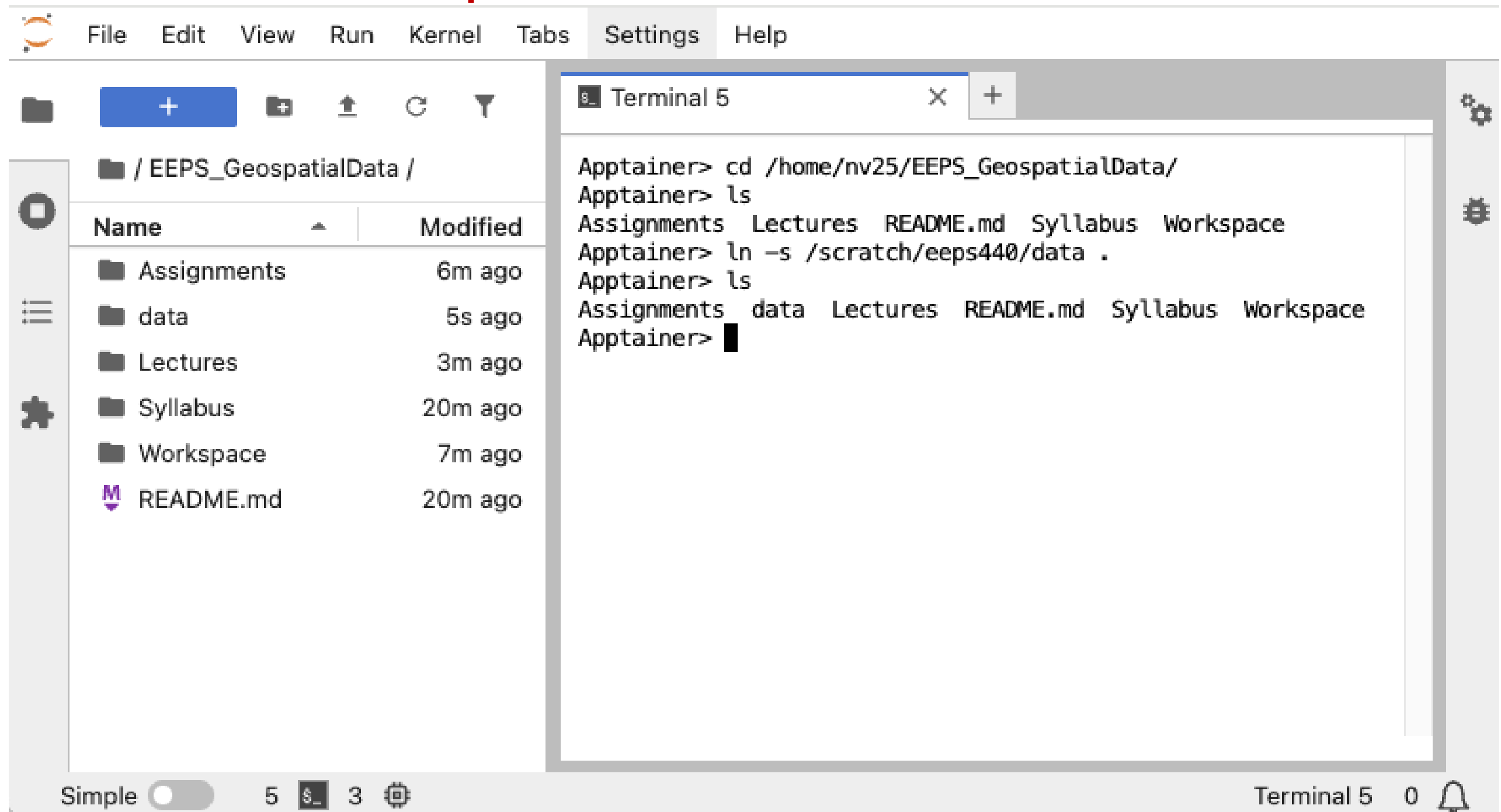
The 'README.md' file content is as follows:

```
1 # EEPS 440/640 - Geospatial Data Science
2
3 ## Course Information
4 - **Course time**: Tu-Th 4-5:15pm
5 - **Course classroom**: KWG 128
6
7 ## Contact Information
8 - **Instructor**: Dr. Noemi Vergopolan
9 - **Office**: KWGL 301
10 - **Email**: nv25@rice.edu
11 - **Office hours**: By reservation ([calendly.com/noemi-vergopolan]
12   ([https://calendly.com/noemi-vergopolan]) Fridays 10am-12pm
13
14 ## Course Objectives and Learning Outcomes
15 This course provides a comprehensive foundation in computational tools and machine
16 learning techniques for exploring geospatial data-driven insights across disciplines.
17 Students will develop key skills in Python programming, data analysis, and machine
18 learning, applying these techniques to large and complex datasets in areas like climate,
19 hydrology, and remote sensing. Through hands-on assignments and projects, students will
20 learn to handle multi-dimensional data, visualize patterns, and perform statistical,
21 spatial, and machine learning analyses to address real-world environmental challenges.
22
23 ## Tentative Course Schedule
24
25 | **Date** | **Topic** | **Software** |
26 | **Assignments** | | |
27 |-----|-----|-----|
28 | 01/14 | Introduction | Jupyter/GitHub/Bash |
29 | 01/16 | Python overview | Python |
30 | 01/21 | Multi-dimensional arrays I | NumPy |
31 | 01/23 | Probability/Statistics | Scipy | HW 0
```

Let's create a symbolic link to our datasets with
(we only need to do this once):

`cd /home/USER/EEPS_GeospatialData`

`ln -s /scratch/eeps440/data .`



The screenshot displays the JupyterLab interface. On the left, the file browser shows the directory `/EEPS_GeospatialData/` with a table of files and folders:

Name	Modified
Assignments	6m ago
data	5s ago
Lectures	3m ago
Syllabus	20m ago
Workspace	7m ago
README.md	20m ago

On the right, the 'Terminal 5' window shows the following commands and output:

```
Apptainer> cd /home/nv25/EEPS_GeospatialData/  
Apptainer> ls  
Assignments Lectures README.md Syllabus Workspace  
Apptainer> ln -s /scratch/eeps440/data .  
Apptainer> ls  
Assignments data Lectures README.md Syllabus Workspace  
Apptainer> 
```

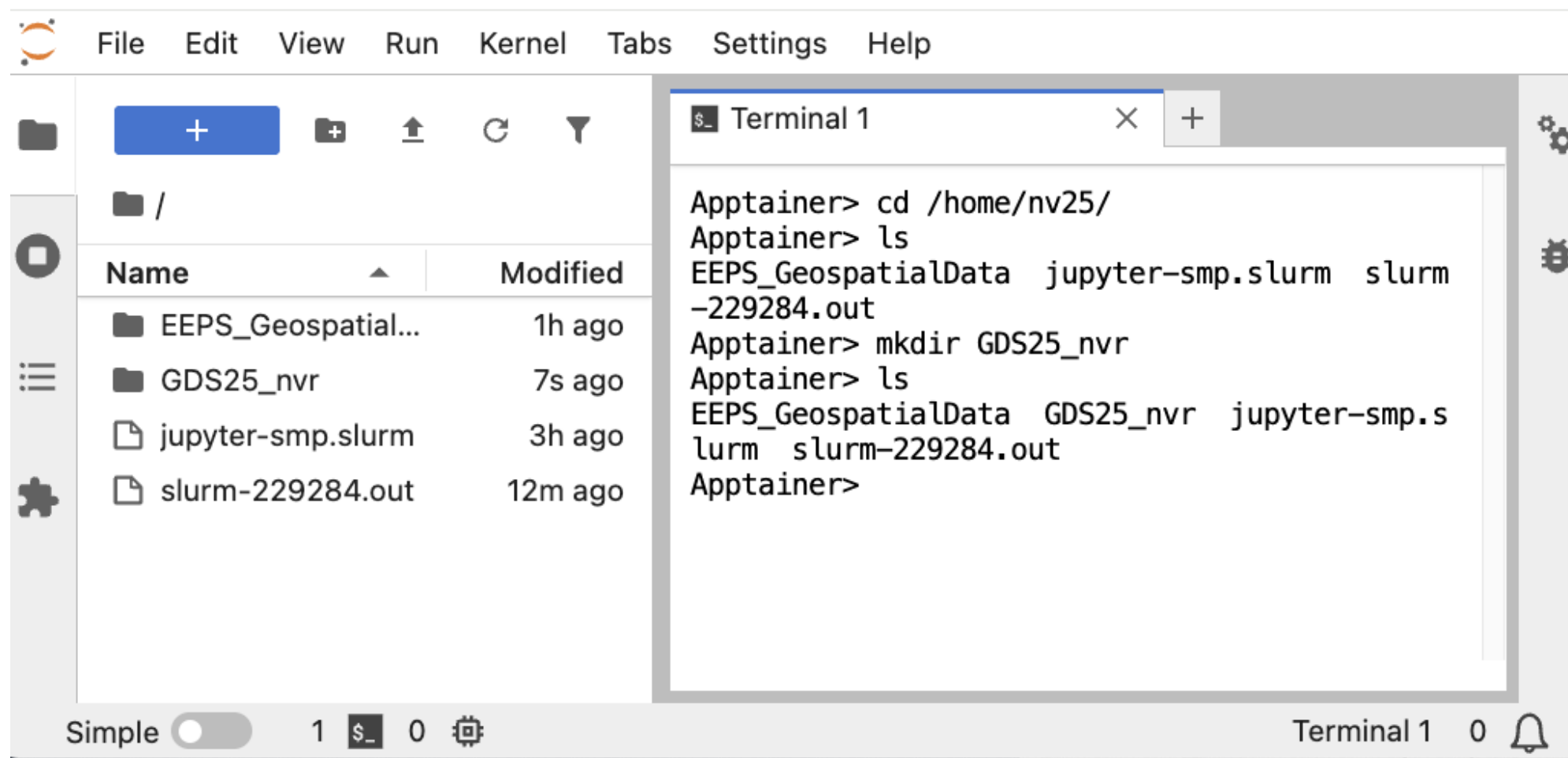
The bottom status bar indicates 'Simple' mode, 5 files, 3 tabs, and the active terminal window 'Terminal 5'.

You will be doing all of your work on
your container.

But how will you access the assignments
and how will you submit them?

Make a new folder to save your assignments

`cd /home/netID`
`mkdir GDS25_Initials`



Create a directory (use your own initials)

Enter directory
`cd GDS25_Initials`

Start the git local repository
(you only need to do this once)
`git init`

Link your folder with your existing repository in GitHub
(you only need to do this once)

`git branch -M main`
`git remote add origin https://github.com/UserName/GDS25_Initials.git`

Copy over HW0 from the coursework folder

```
cp ../EEPS_GeospatialData/Assignments/HW0.ipynb .
```

Add the assignment to your local repository

```
git add HW0.ipynb
```

Commit HW changes to your local repository

```
git commit -m 'add HW0 file'
```

Push your HW from local repository to your GitHub repository

```
git push -u origin main
```

GitHub UserName

Token Password

Now they are attached to each other!

And your GitHub version has been updated.

 **GDS25_nvr** Private

Unwatch 1

Fork 0


Star 0


main

Go to file


+

Code

 **Noemi Vergopolan Da Rocha** add HW0 file a2e2e4a · 22 minutes ago

 HW0.ipynb add HW0 file 22 minutes ago

README



Add a README

Add a README with an overview of your project.

Add a README

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

With every new assignment:

- Go to your assignments folder
- Copy the assignment from the coursework repository
- Add it to your local repository (git add)
- Save and commit the changes (git commit)
- Complete the assignment
- Save and commit the changes (git commit)
- Push the changes to the online repository (git push)

- But why so complicated? Why not just send the completed assignment via email?
- Because using version control is critical to most research and industry data science nowadays.
- Forcing you to use version control throughout the course will ensure you learn how to use it.

Assignments submitted any other way will NOT be accepted.

HW0

Due February 6 (before class)

Get your container up and running, clone the class repository, create and link your private repository, and solve some introductory Python exercises, push your HW to your repository.

This will be hard for many, but if we get this out of the way at the beginning, we will avoid a lot of headaches moving forward.