

CEE 690-02

ENVIRONMENTAL SPATIAL
DATA ANALYSIS (ESDA)

Lecture I

Introduction

- Instructor: Nathaniel Chaney (Nate)
- TA:TBD

Motivation

There are big environmental challenges that we want to take on...

Here are some examples from Hydrology

Floods

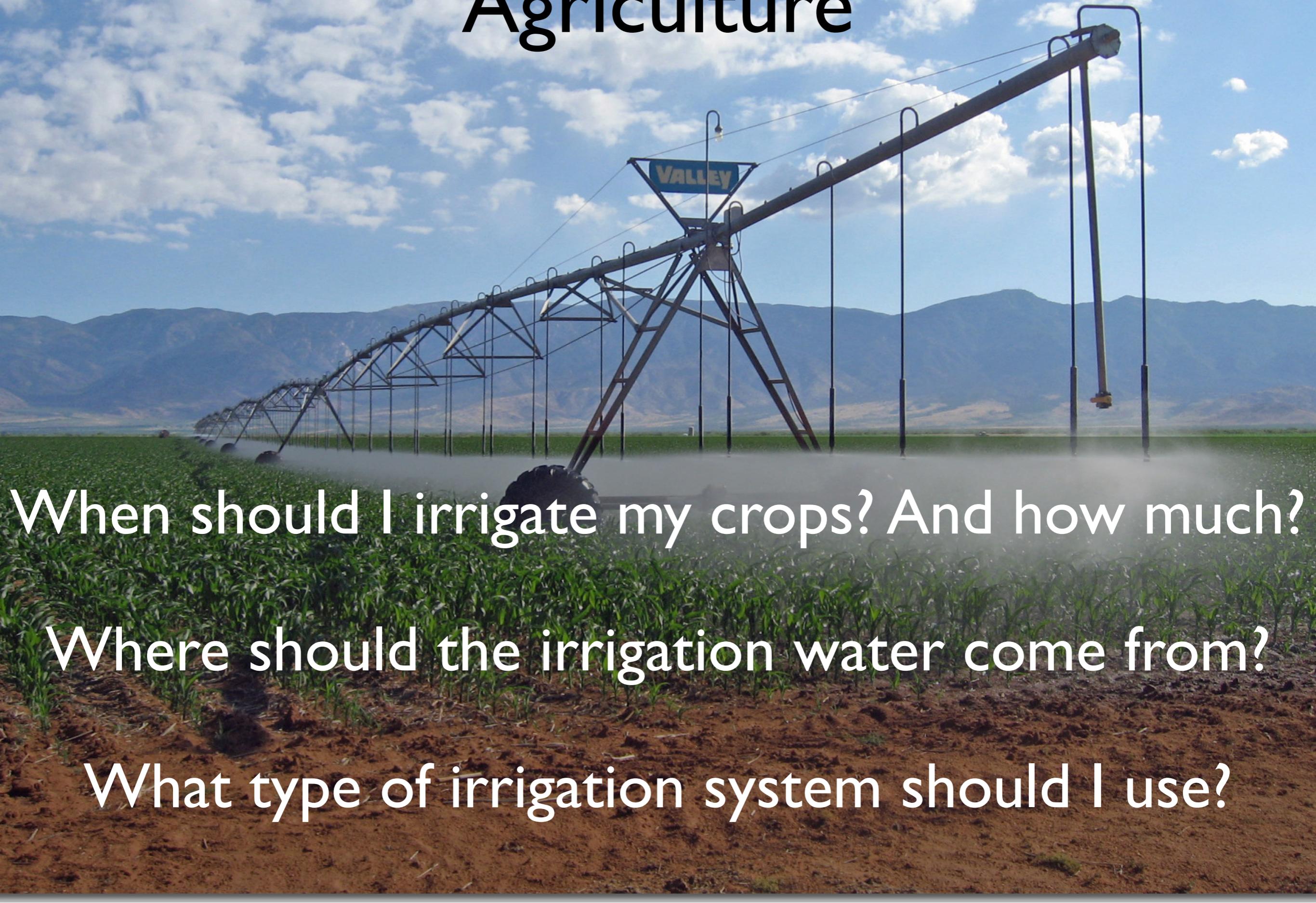


Where will it flood during a hurricane?

How can we mitigate the impacts of floods?

What is the risk that my house is flooded?

Agriculture



When should I irrigate my crops? And how much?

Where should the irrigation water come from?

What type of irrigation system should I use?

Water management

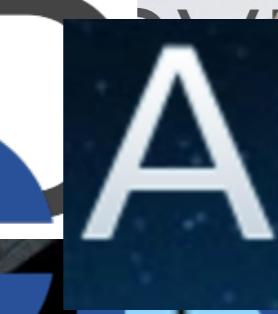
Where should I place a reservoir?

How much water should I release?

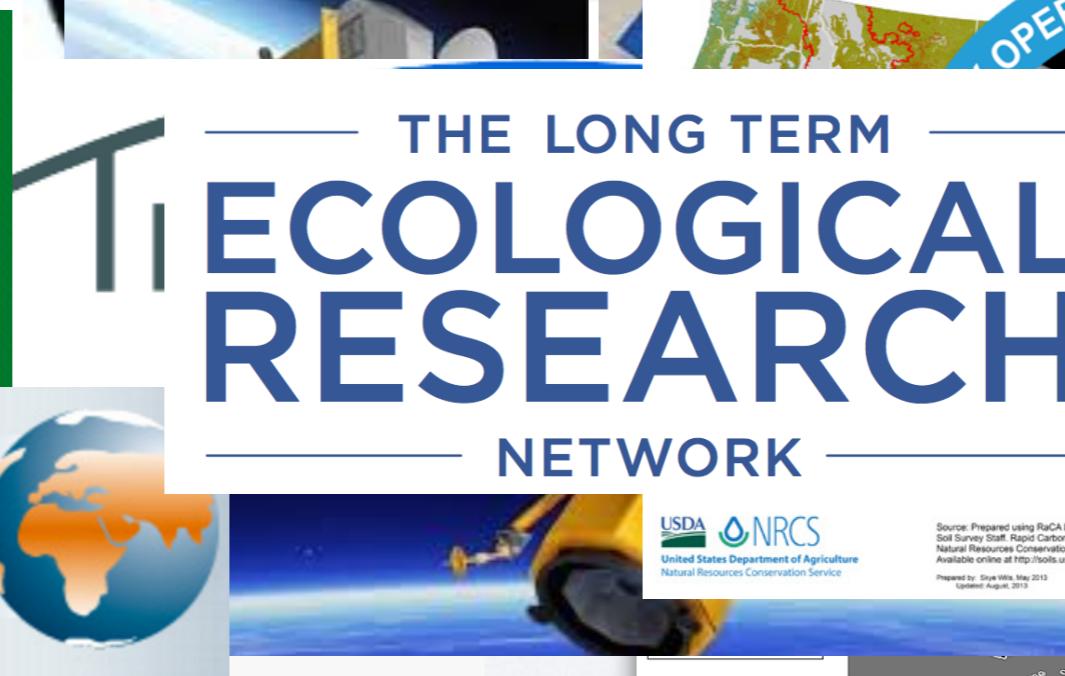
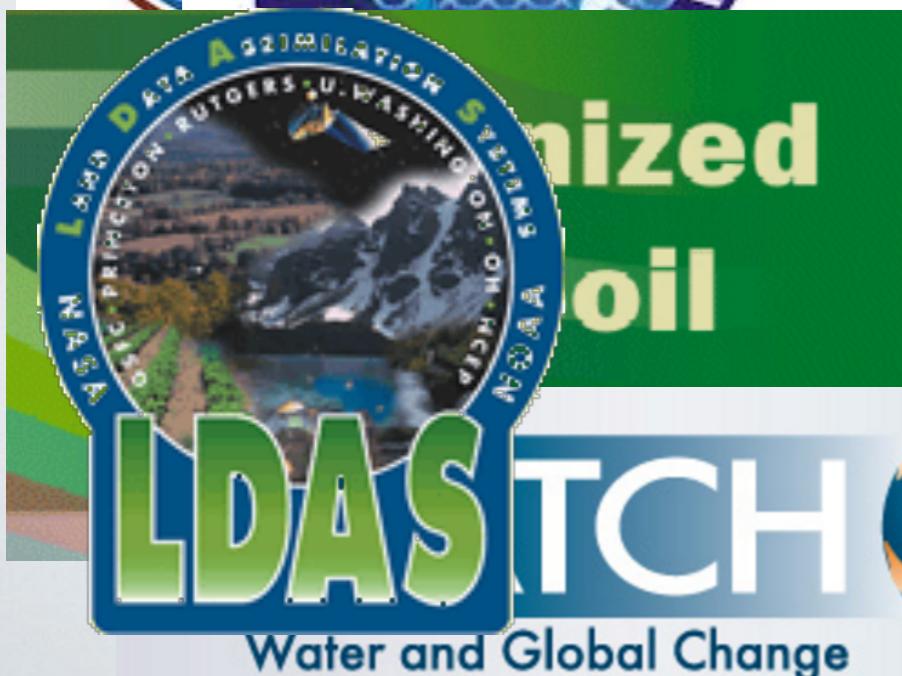
How will a changing climate impact water inflows
into existing reservoirs?

How do we address those
questions?

Data, data, and more data



“Age of Data”

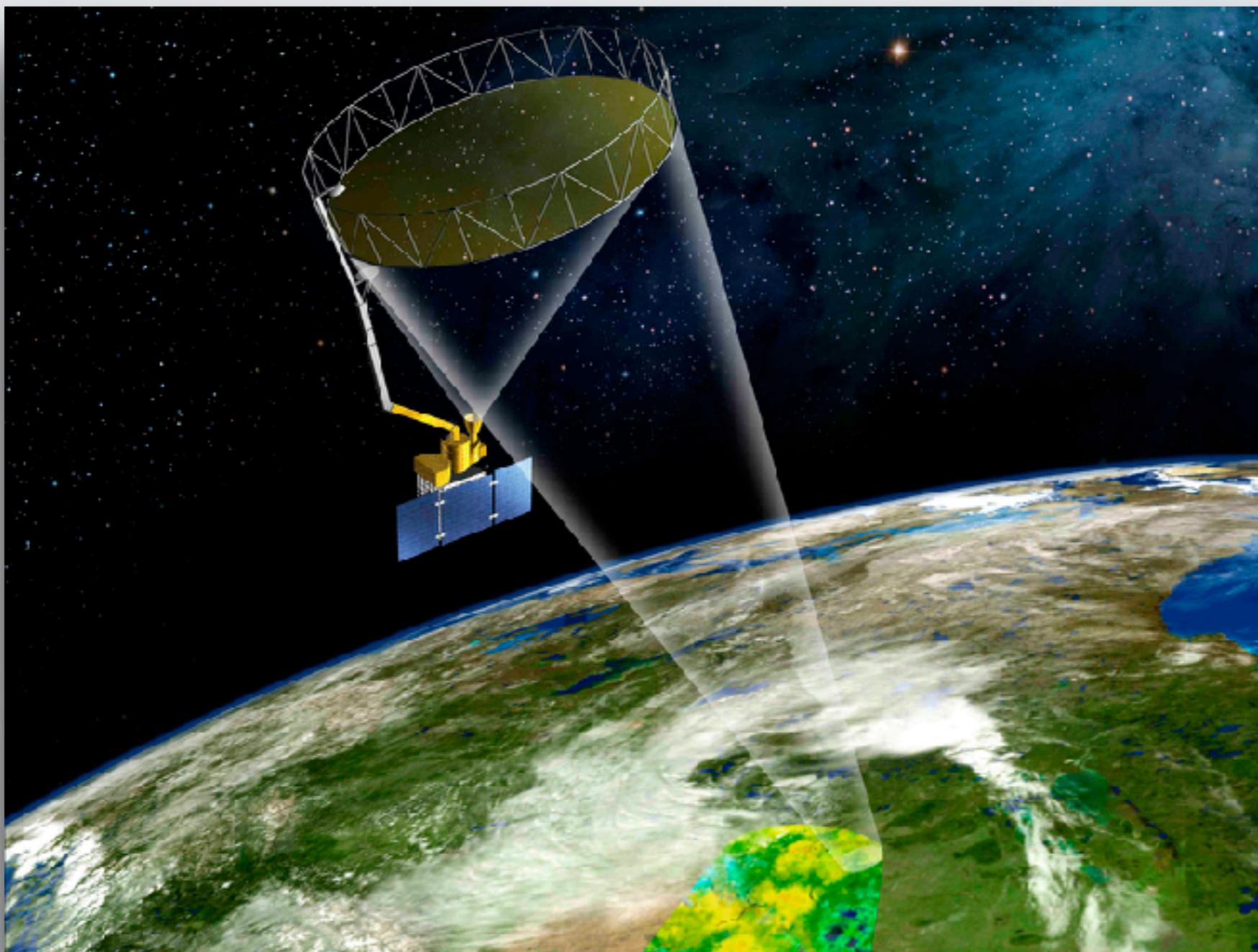


What Petabytes of data feels like

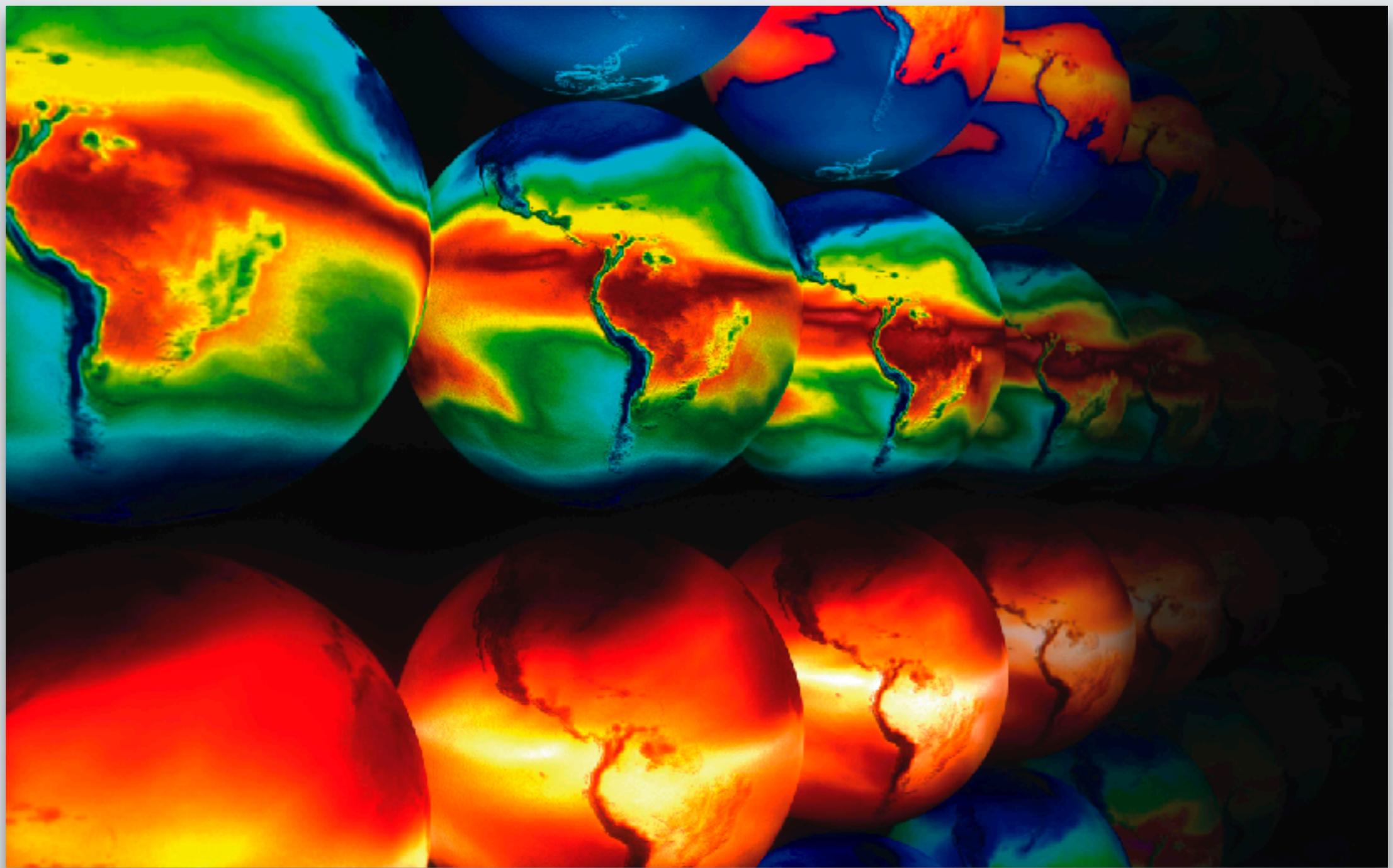


And the problem is even much
worse with spatial data

Example: Satellite remote sensing



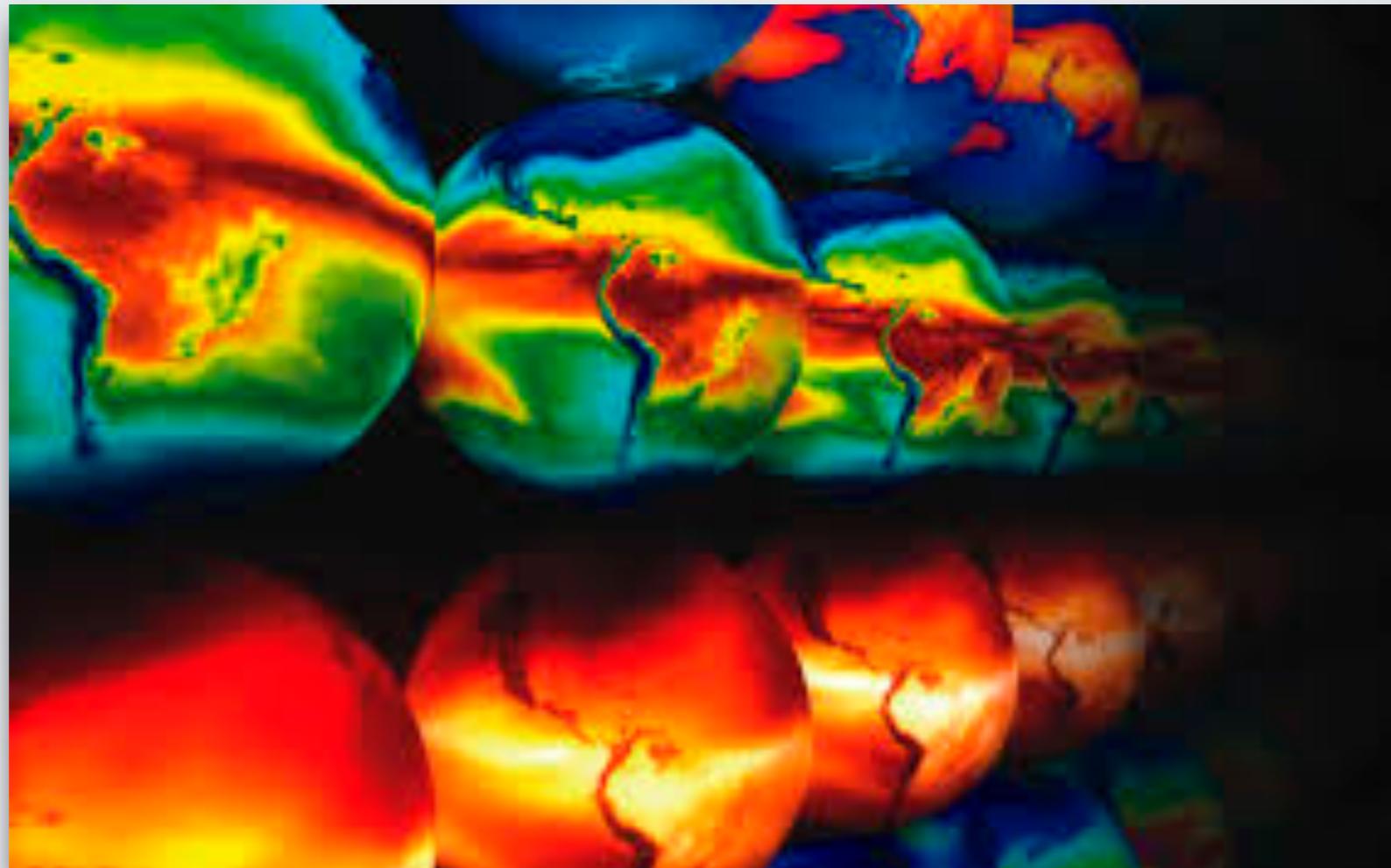
Example: Climate model output



This course aims to give you the
tools and knowledge on how to
tackle these data

What topics will we cover?

Topics: Analyze spatial environmental data



How can we analyze terabytes/petabytes of climate model output and/or satellite data?

Topics: Map projections



Robinson

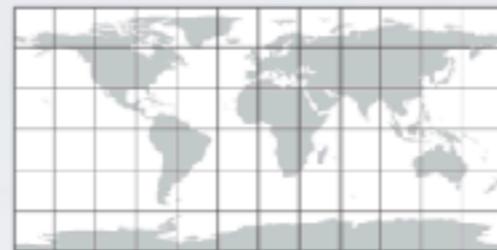


Plate Carrée



Winkel Tripel



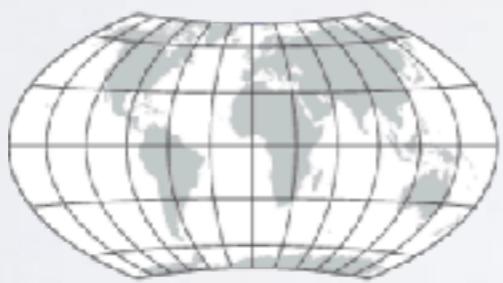
Eckert IV



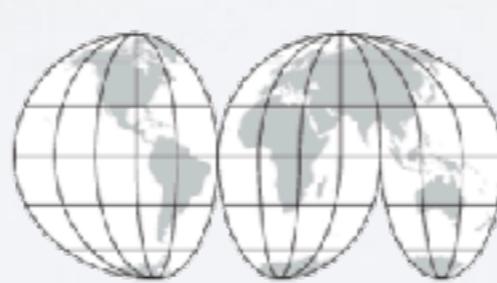
Mollweide



Mercator



Wagner VII



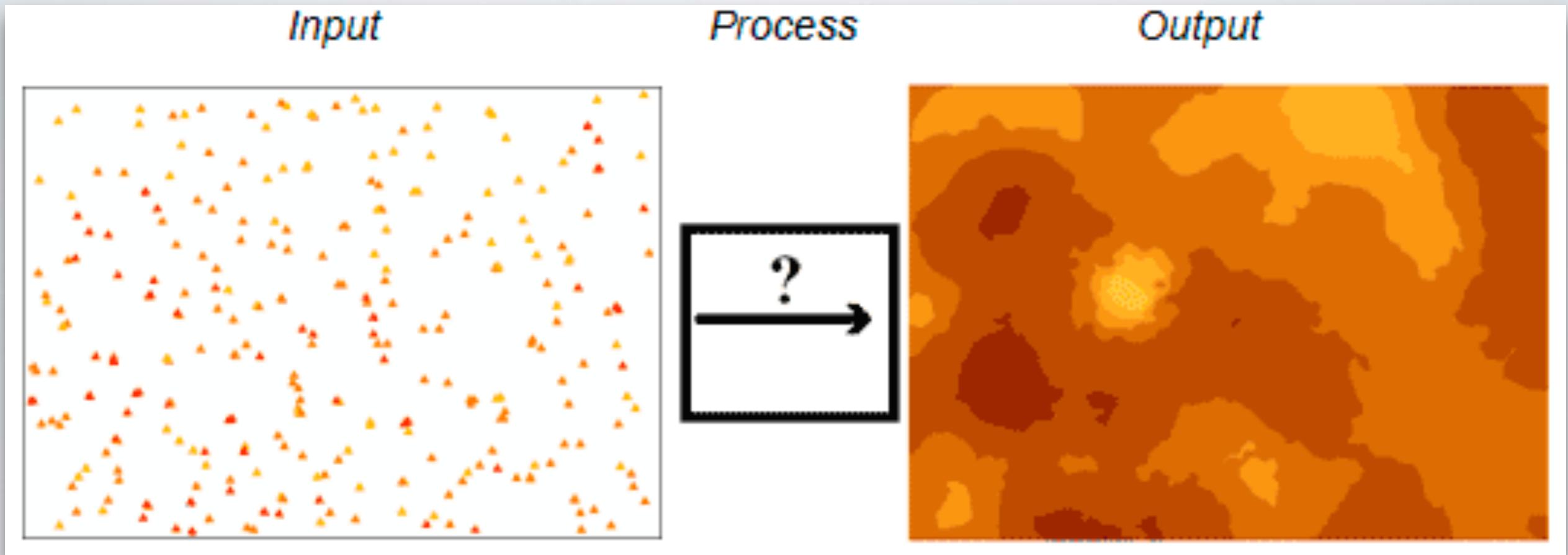
Interrupted Mollweide



Goode Homolosine

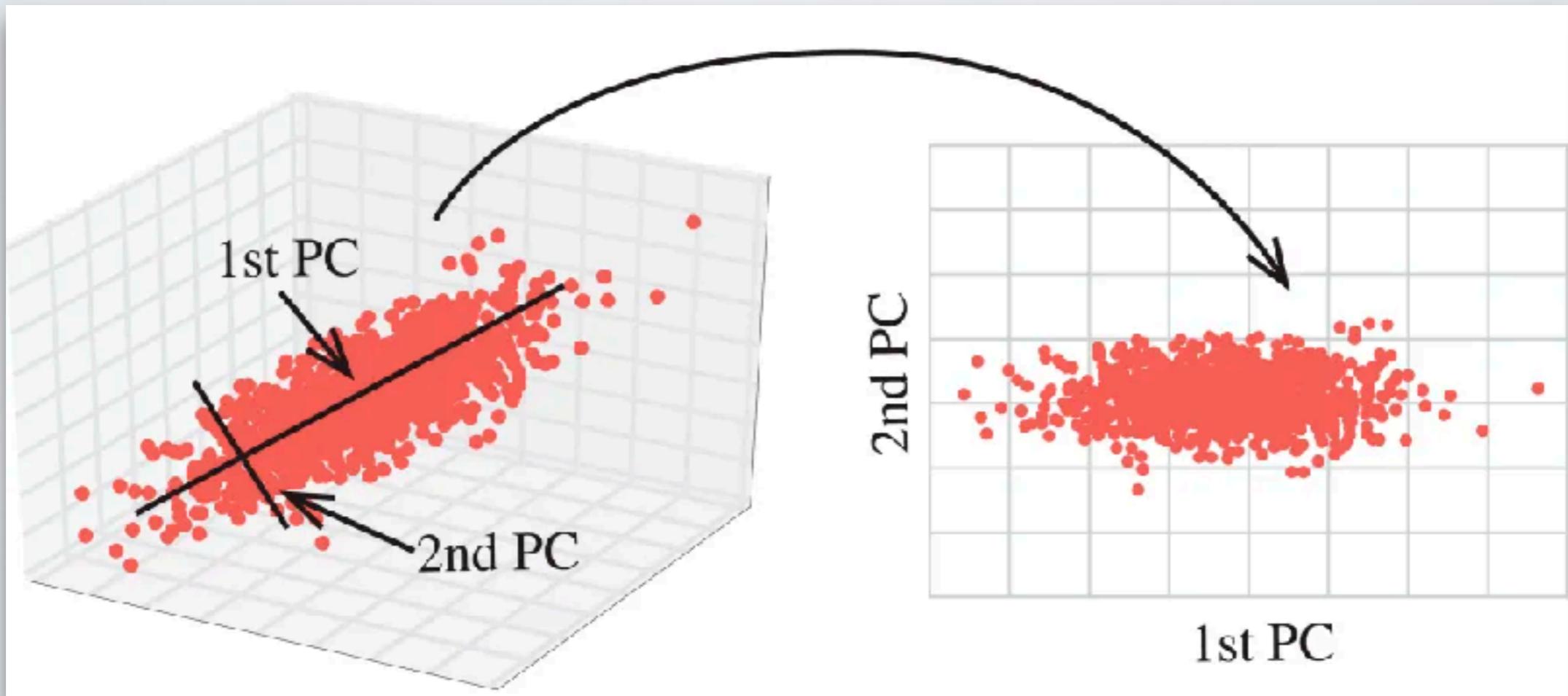
How can we transform a map to a different projection?

Topics: Spatial interpolation/ prediction



How do you create a spatial map from point data?

Topics: Reduce dimensions

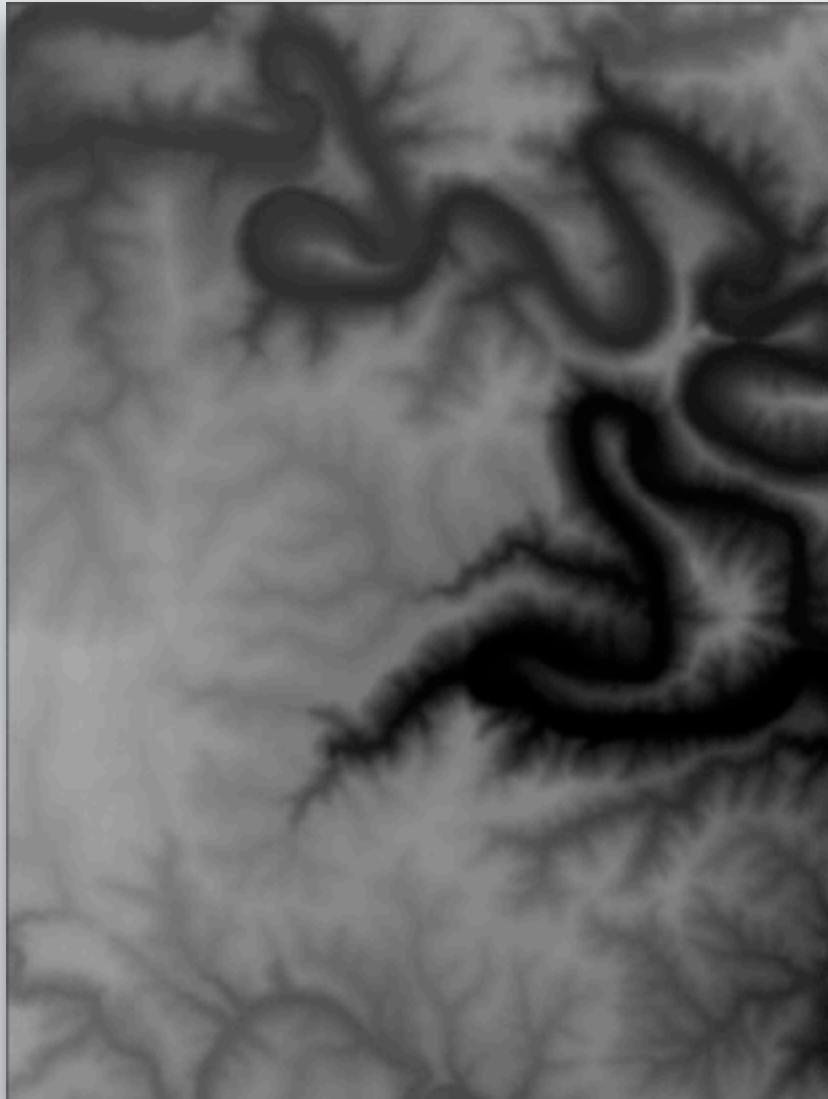


How do we compress data?

What are the most important features in the data?

Topics: Terrain analysis

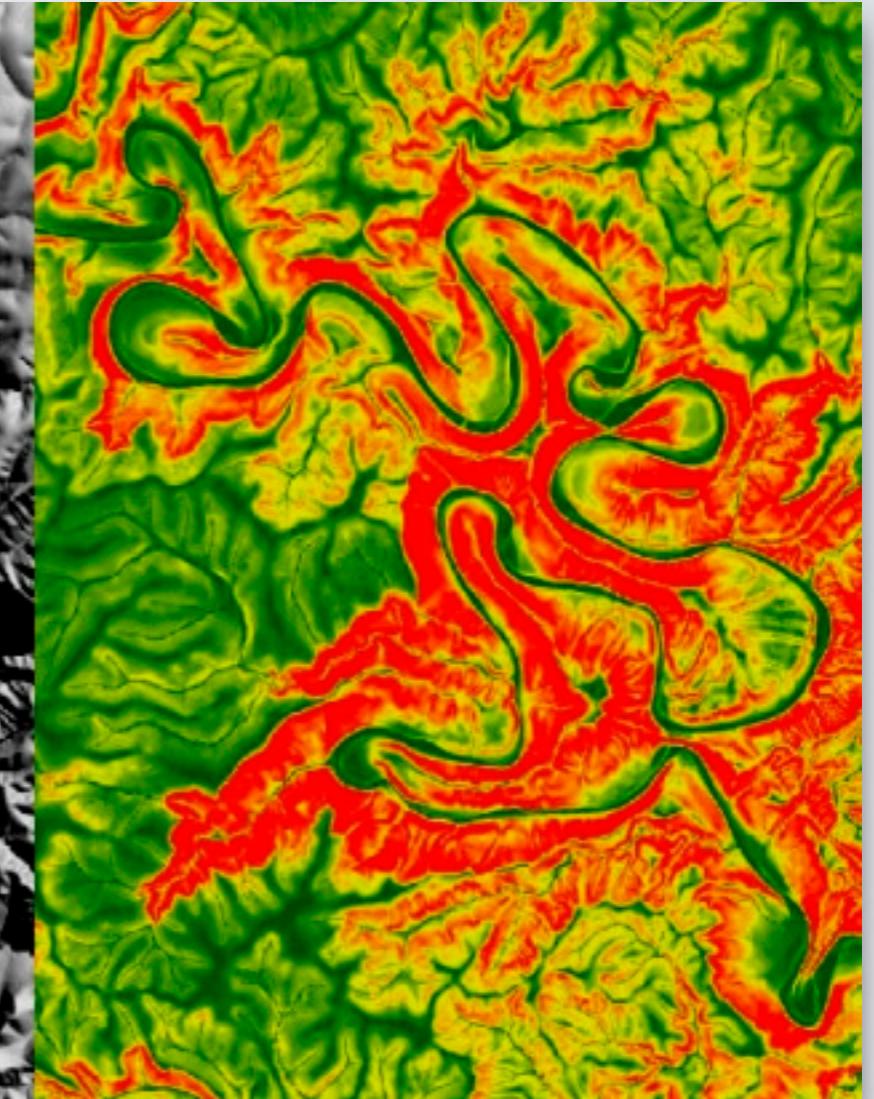
Elevation



Hillshading



Slope



What can we learn from elevation data?

And more...

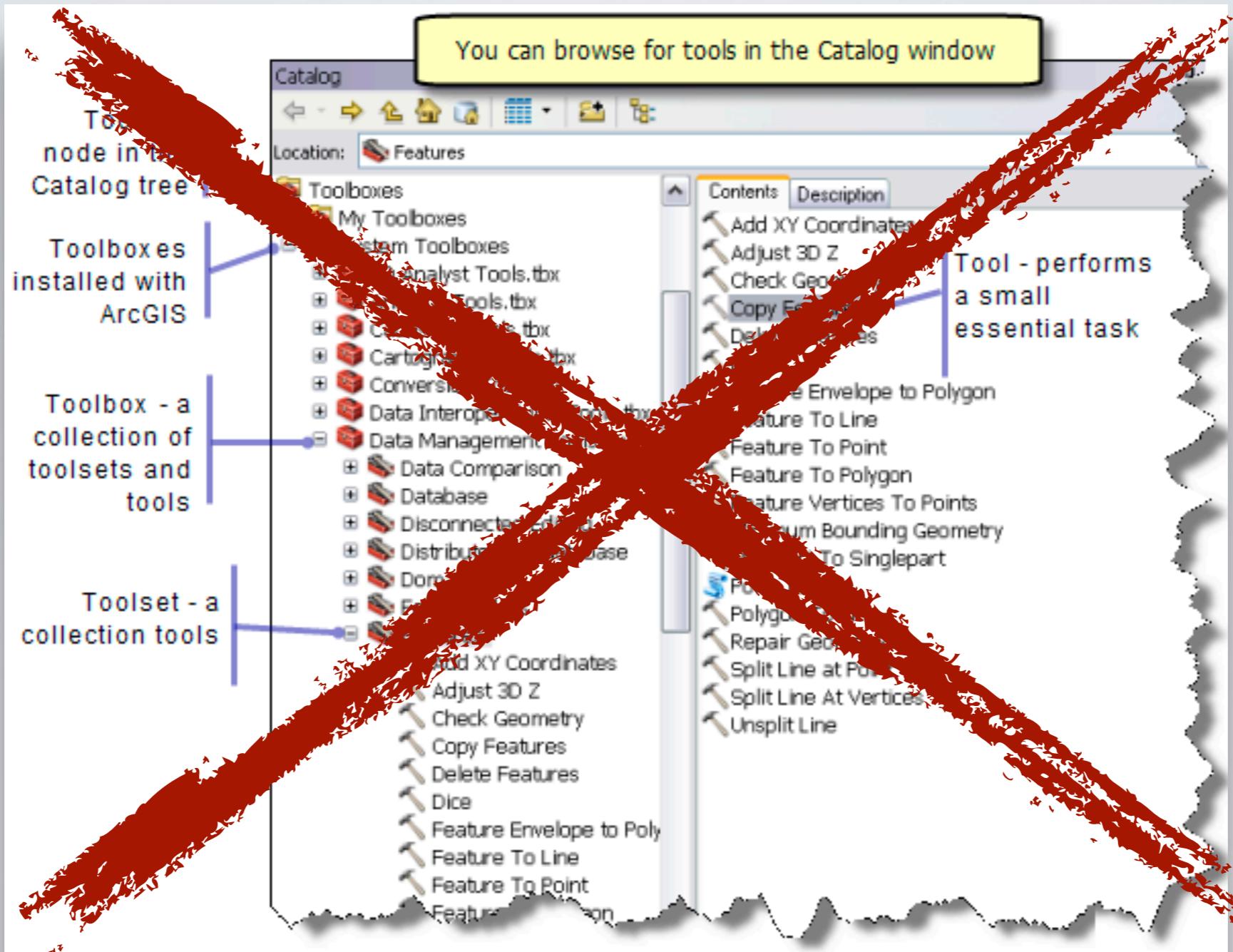
But what tools will we use to
learn how to create and analyze
these large datasets?

Let's start off with what we will
NOT be using this semester

This is NOT an ArcGIS course



This is NOT a “click the button” class



If that was what you were expecting then
this may not be the class for you

Then what will we use?

Programming, programming, and
more programming

If you are terrified of programming... Don't
be! It is worth the time investment.

However, if you are not willing to dedicate
the time and just want to “pass by” then
this class is probably not a good fit.

Which programming languages
will we use?

Primary language: Python



Python level: Intermediate

Secondary language: Bash



Bash level: Beginner

Don't be afraid...

- We are going to spend the first third of the class in learning how to program in Python (and Bash)
- It will get hard but it will be additive so one can take this class without a background in these programming languages

Syllabus

Instructor

Professor Nathaniel W. Chaney (Nate)

Email: nathaniel.chaney@duke.edu

Office: FCIEMAS 2463

Office hours: Thursdays after class via Zoom

TA

TBD

Email: TBD

Office: TBD

Office hours: TBD

Course Logistics: General Info

- Grading:
 - Homework: 40%
 - Final Project: 40%
 - Participation: 20%

Homework

- Completed assignments will be submitted via each student's private GitHub repository
- Each assignment must be submitted before class on the day listed on the schedule
- Late homework will not be accepted.

Textbooks?

There are none.

Welcome to the wild, wild west!

We are here to help. That being said,
Google will answer almost all questions
much faster than we will (Use it!).

Let me
Google
that for you.

Participation I

- Discussion/questions - Avoid the one-way lecture approach relies on your involvement
- Bring you laptop to class to follow along with the interactive lectures



Final Project

The final project can be done in groups of 2 or individually. The expectations for the project will increase with the group size. It will involve the following components:

- Initial Proposal (October 22nd via email)
 - 3 pages max, single-spaced, 12 point font size, 1 inch margin
 - Contains: Title, introduction, objectives, data, methodology, and timeline of tasks
- Oral presentation (November 17th and 19th in class)
 - 12 minute oral presentation, 3 minutes for questions
 - Everyone needs to be present for each presentation
- Final report (November 24th via email)
 - 10 pages max, single-spaced, 12 point font size, 1 inch margin
 - Contains: Title, introduction, data, methods, results, discussion, and conclusion

Tentative schedule I

Date	Topic	New Software	Assignments	Presentation
08/18	Introduction	Jupyter/GitHub/Bash	-	-
08/20	Python overview	Python	-	-
08/25	Multi-dimensional arrays I	NumPy	-	TBD
08/27	Visualizing data	Matplotlib	-	TBD
09/01	Data storage	Pickle/H5py/NetCDF/Tiff	-	TBD
09/03	Probability/Statistics I	Scipy	-	TBD
09/08	Probability/Statistics II	-	HW #1 due	-
09/10	Time series analysis	-	-	TBD
09/15	Map projections I	Cartopy	-	TBD
09/17	Map projections II	GDAL	-	TBD
09/22	Multi-dimensional arrays II/Downloading data	Xarray	-	TBD
09/24	Vector Data	OGR/Shapely/Fiona	-	TBD

Tentative schedule II

09/29	Cluster Analysis I	Scikit-Learn	-	TBD
10/01	Cluster Analysis II	-	-	TBD
10/06	Dimensionality Reduction	-	HW #2 due	-
10/08	Classification/Regression I	-	-	TBD
10/13	Classification/Regression II	-	-	TBD
10/15	Classification/Regression III	-	-	TBD
10/20	Classification/Regression IV	-	-	TBD
10/22	Geostatistics I	-	Proposal due	TBD
10/27	Geostatistics II	-	HW #3 due	-
10/29	Geostatistics III	-	-	TBD
11/03	Geostatistics IV	-	-	TBD
11/05	Terrain Analysis I	-	-	TBD
11/10	Terrain Analysis II	-	-	TBD
11/12	Scaling up code	Numba/Mpi4py/Dask	-	TBD
11/17	Oral Presentations	-	-	-
11/19	Oral Presentations	-	-	-
11/24	Written report due	-	HW #4 due	-

Docker container

All work in this class will be done online on your own personal Docker container

The screenshot shows a web browser window with the URL vm-manage.oit.duke.edu in the address bar. The page has a dark blue header with the Duke University logo and the text "Duke | VM Manage". On the right side of the header are links for "Home", "Docker", and "Logout". The main content area is white. On the left, there's a section titled "About VM-Manage" with text about the service and a link to the Virtual Computing Manager. On the right, there's a section titled "Docker containers" with text about reserving personal computer environments and a link to access Docker containers. At the bottom, there's a footer with the Duke University logo, copyright information, and links to Accessibility, Style Guide, About Duke.edu, Contact Us, and social media icons.

About VM-Manage

VM-Manage is a reservation service for compute environments such as RStudio, Eclipse, Jupyter, Spyder+Anaconda Python and other course-oriented Docker containers.

For virtual machine reservations visit the new **Virtual Computing Manager** service at: <https://vcm.duke.edu>

Docker containers

Students and instructors can reserve personal computer environments running applications such as RStudio, the Eclipse IDE, Jupyter Notebooks, Matlab, Octave, Mathematica, and others for a semester.

[Click here](#) to access your personal Docker containers.

Duke
UNIVERSITY

© 2012-2016 Duke University, Durham, NC 27708 | [Accessibility](#) | [Style Guide](#) | [About Duke.edu](#) | [Contact Us](#) |

Look for CEE690-02

Docker Containers

Welcome back Nathaniel W Chaney!

Your reserved application containers:

KerasTensorflow - Jupyter with Keras and Tensorflow-GPU

Your KerasTensorflow reservation has expired - container reservations are valid for one semester.

[Click here to renew your reservation](#)

MATLAB - mathematics and matrix manipulations

Your MATLAB reservation has expired - container reservations are valid for one semester.

[Click here to renew your reservation](#)

CEE690 - Environmental Spatial Data Analysis

[Click here to log in](#) to your CEE690 environment

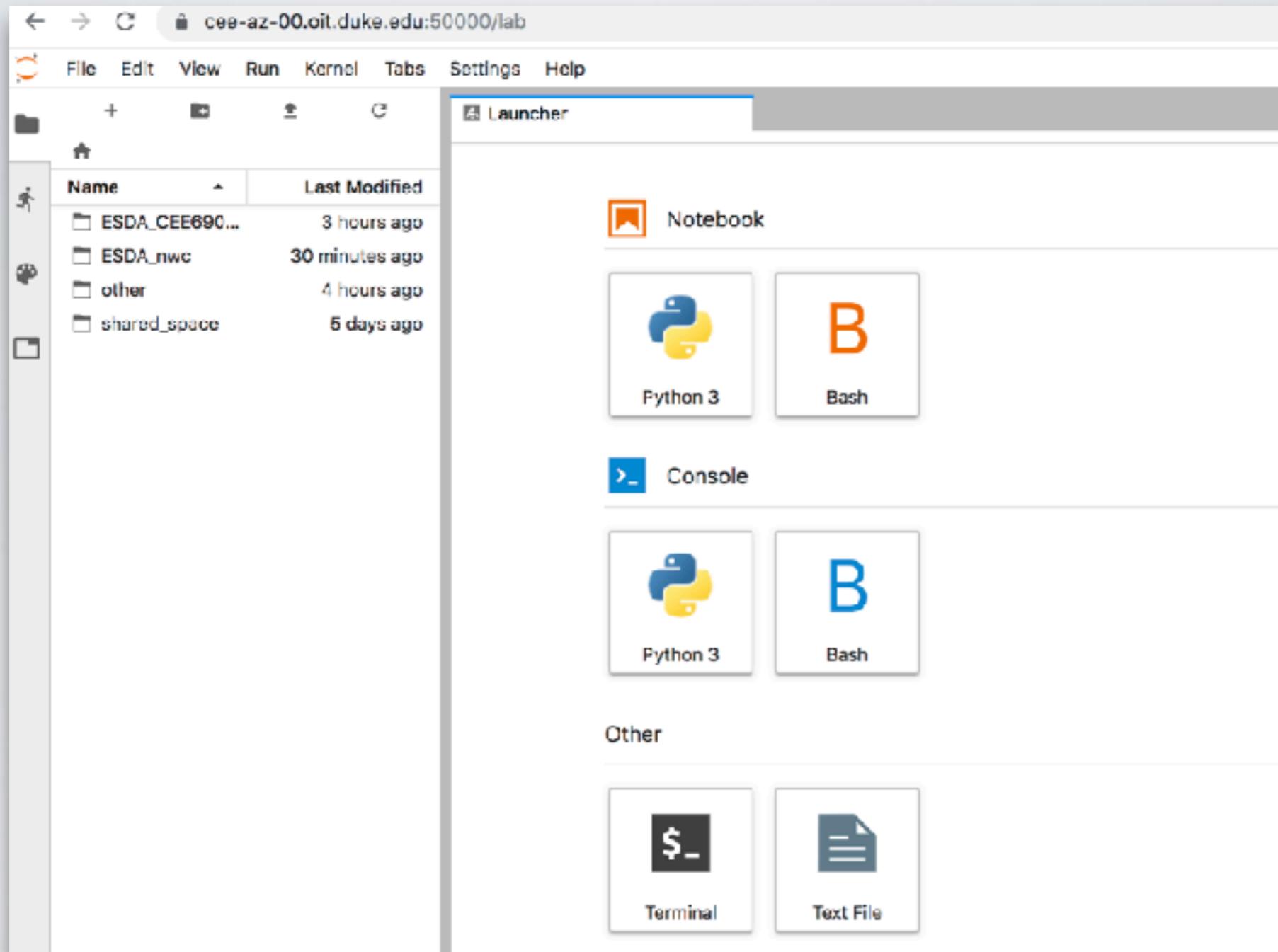
If you are unable to connect to your CEE690 environment you can [request a restart](#).

Spark - PySpark notebooks for Apache Spark cluster

Your Spark reservation has expired - container reservations are valid for one semester.

[Click here to renew your reservation](#)

Jupyter Lab

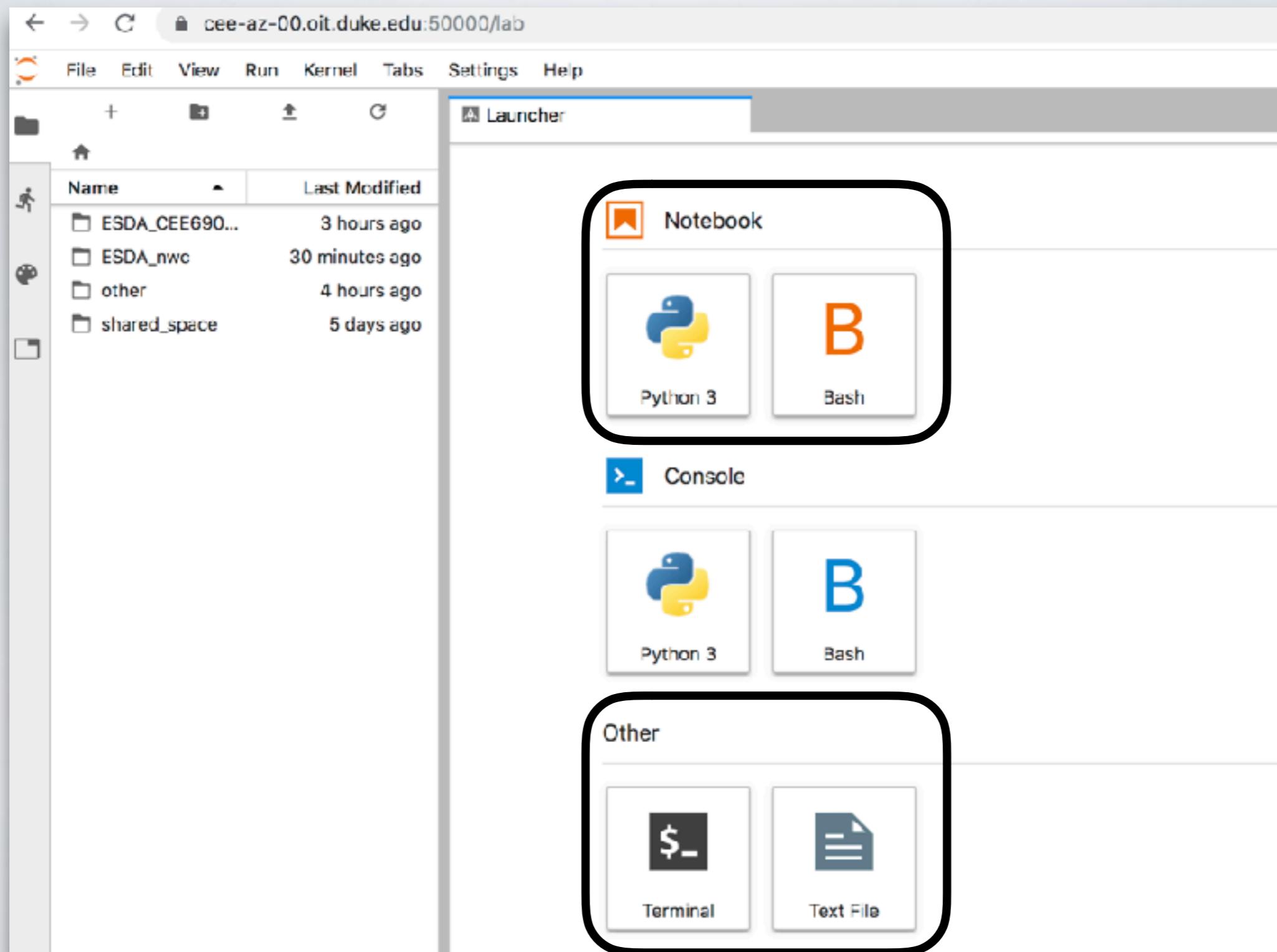


<https://jupyterlab.readthedocs.io/en/stable/>

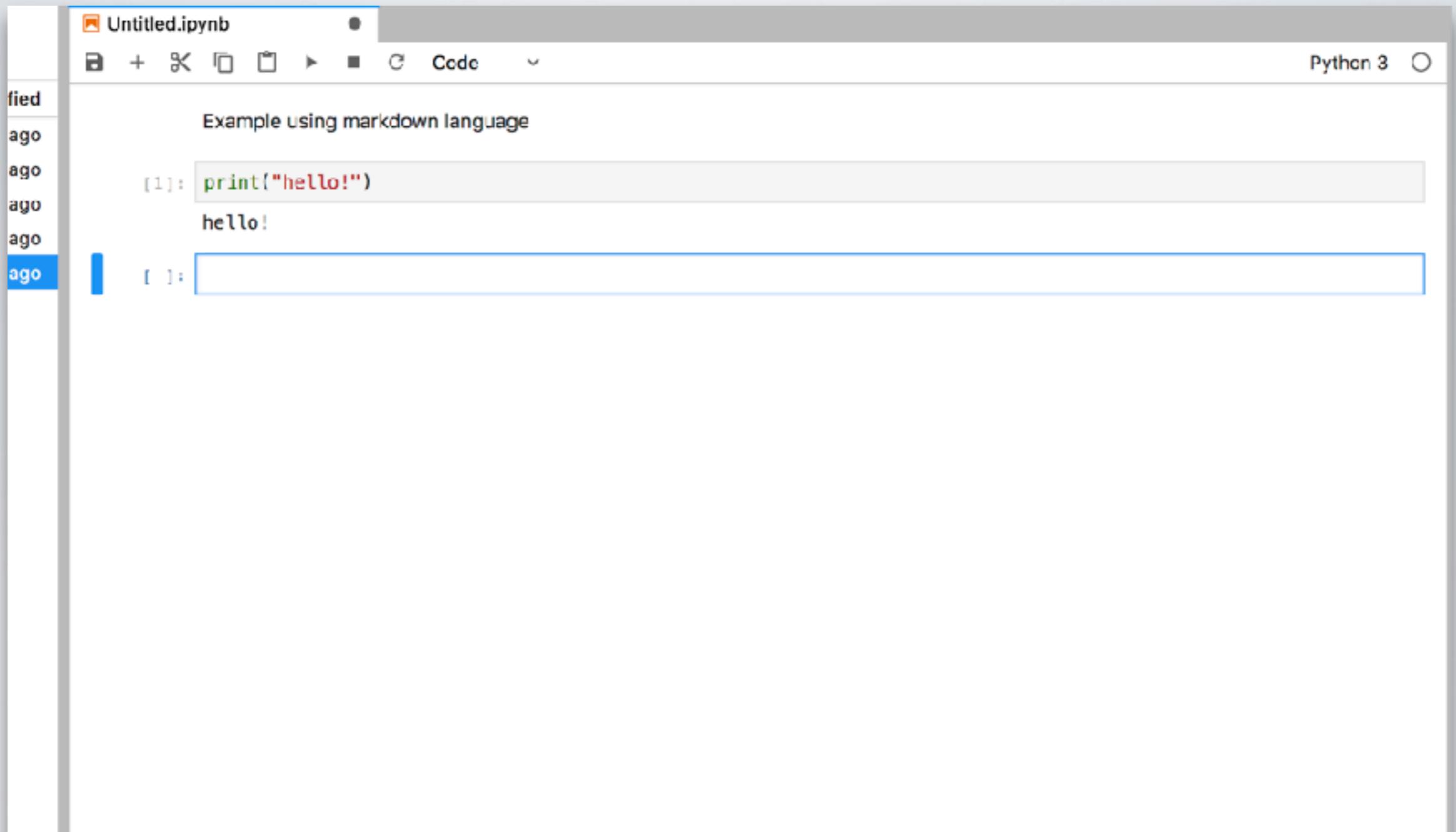
A way to understand this...

- The Docker container is your own computer (virtual machine)
- Jupyter lab is a way to interact with that computer.
- This is the easiest way to do this. There are certainly more complex ways (and more powerful) to connect to a virtual computer but for this course we will only use Jupyter lab (and Jupyter notebooks).

Components in Jupyter Lab



Jupyter Notebook



A screenshot of a Jupyter Notebook interface. The title bar shows "Untitled.ipynb". The toolbar includes icons for file operations like new, open, save, and run, along with a "Code" dropdown and a "Python 3" kernel selector. The main area displays a cell containing a markdown example and a code cell with an output. The code cell contains the Python command `print("hello!")`, which has been executed and printed the word "hello!". A new cell input field is visible at the bottom.

```
Example using markdown language
[1]: print("hello!")
hello!
[ ]:
```

https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

Jupyter environment (older view)



You can do the same thing with the older view as with Jupyter lab. Jupyter lab is just more convenient and “prettier”. We will need to use the older view many times though since Jupyter Lab doesn't have some functionality that the older view does.

Terminal



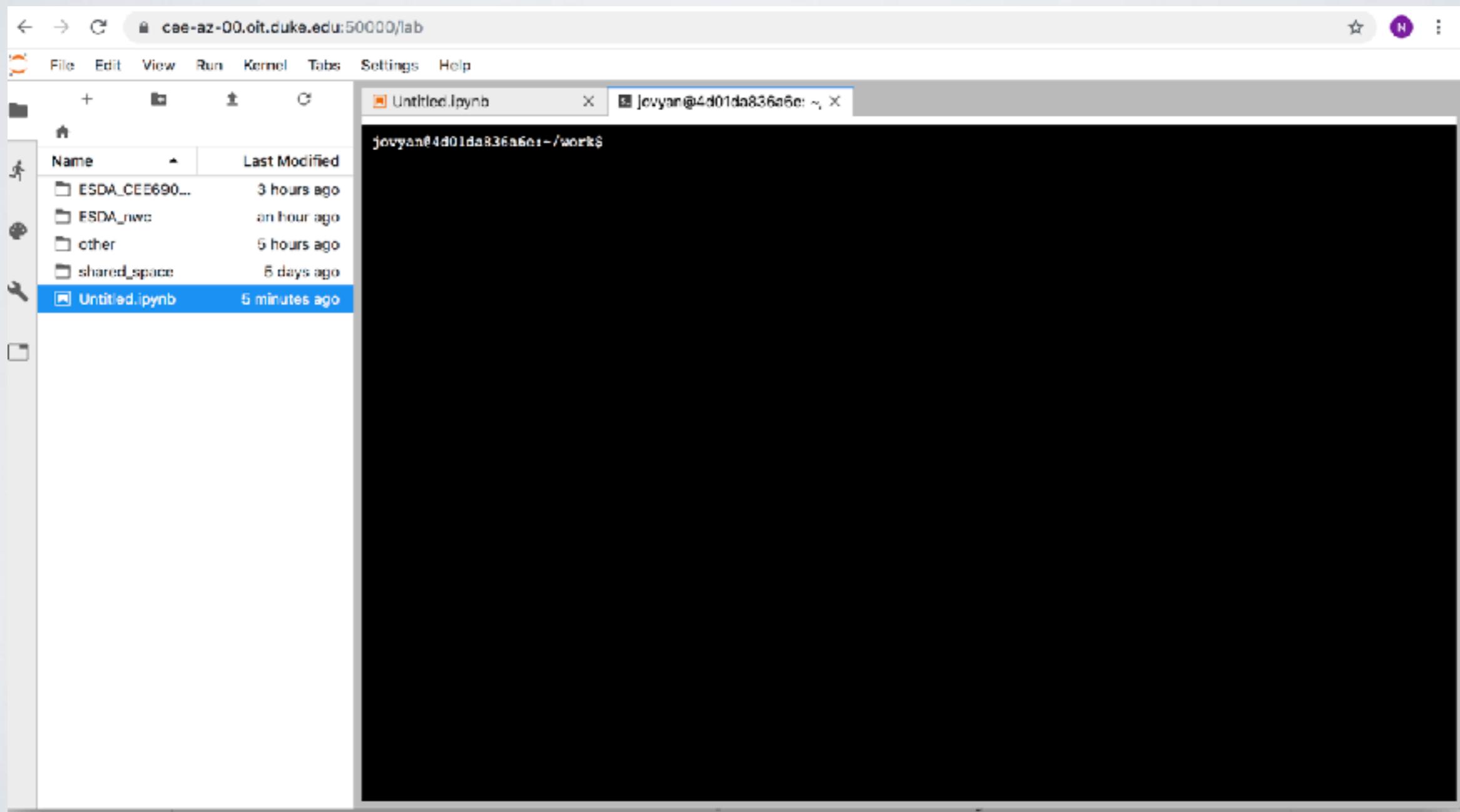
Remains one of the best ways to interact more directly with the computer (flexibility!)

Linux Ubuntu: The Docker container's operating system under the hood



So you will actually be using Ubuntu throughout the semester (you just won't really even notice)

Terminal in Jupyter lab (i.e., Ubuntu terminal)



Interacting with the terminal: Bash (Bourne Again SHell)



Tutorials to learn Bash

<https://linuxconfig.org/bash-scripting-tutorial-for-beginners>

<https://www.codecademy.com/learn/learn-the-command-line>

<https://devhints.io/bash>

<https://hackernoon.com/top-10-bash-file-system-commands-you-can't-live-without-4cd937bd7df1>

Top bash commands

- 1) **ls** - Lists the folder and file names in the current working directory.
- 2) **cd** - Change Directory to the path specified, for example cd projects
- 3) **mkdir** - Make directories with this command mkdir my_folder
- 4) **mv** - Moves files and folders. The first argument is the file you want to move, and the second is the location to move it to.
- 5) **cp** - Copies files and folders cp my_file ./projects . The flag -r recursively copies subfolders and files.
- 6) **rm** - Removes files and folders rm my_folder . Using -r will again recursively delete subfolders, -f force deletes, and -rf for a recursive force delete.

You will be doing all of your work on your Docker container. But how will you access the assignments and how will you submit them?

Let's talk about version control...

GitHub: Git on the web



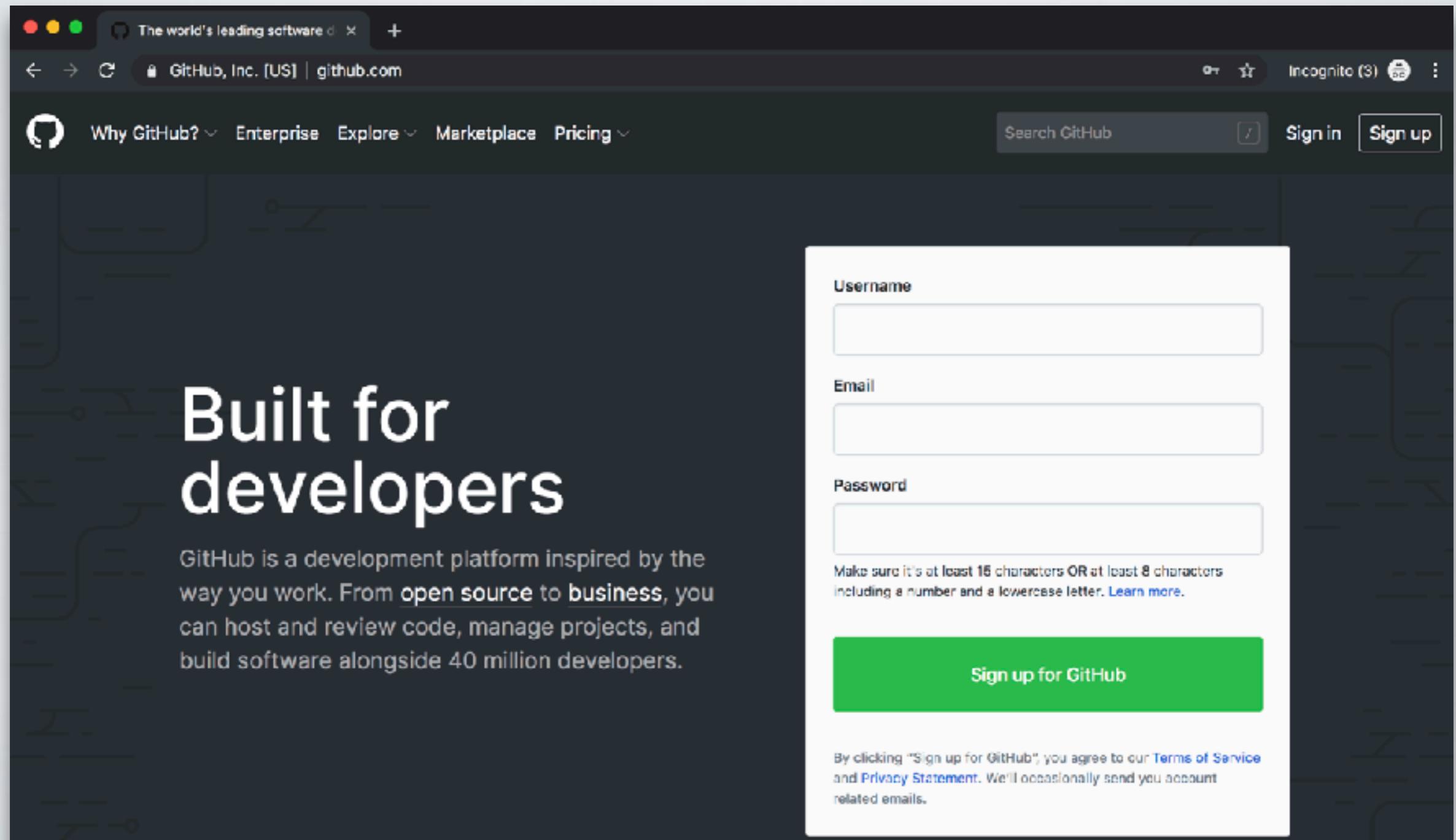
<https://www.youtube.com/watch?v=w3jLJU7DT5E>

Version control: Git

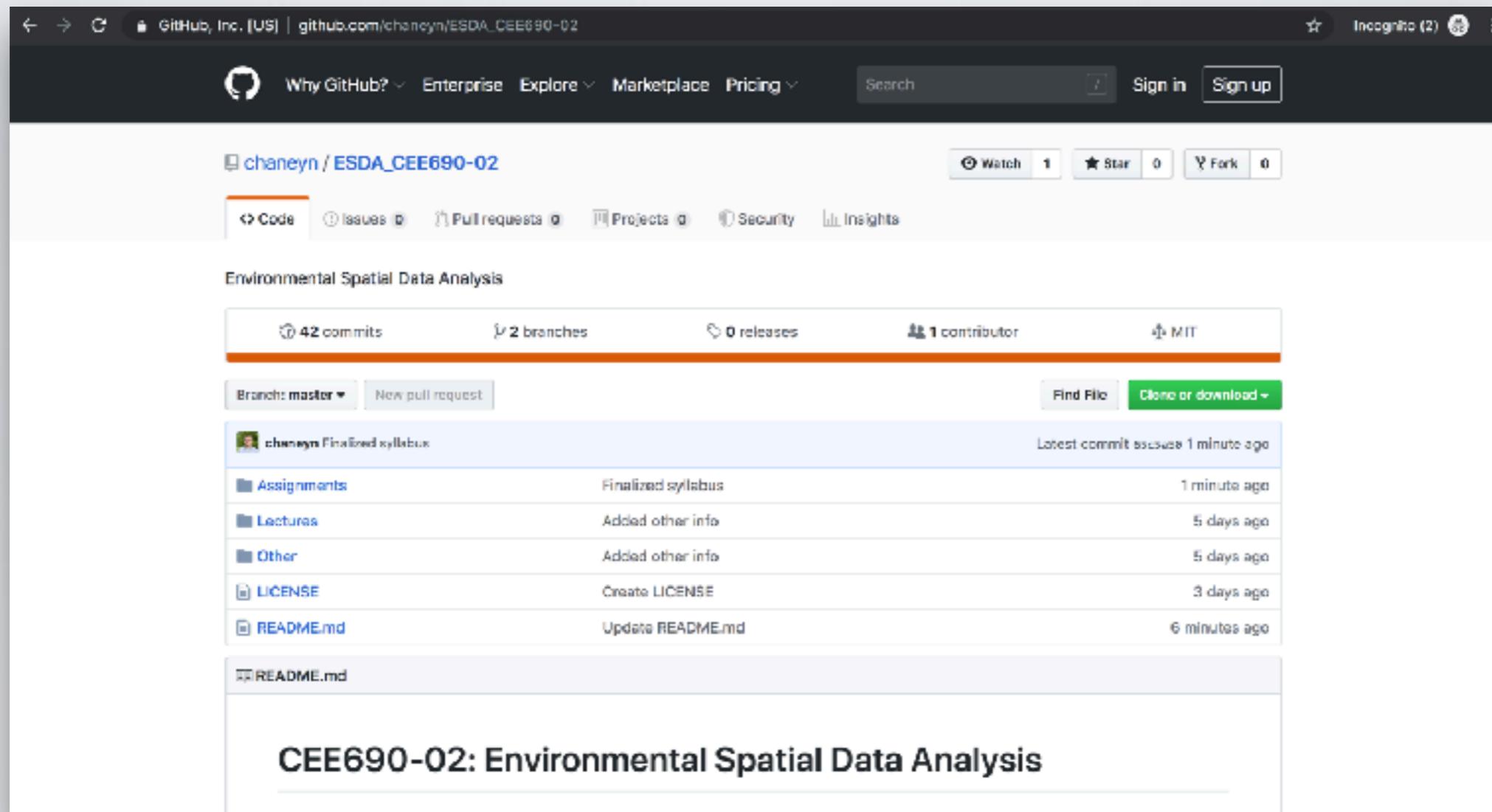


<https://vimeo.com/41027679>

Create a personal account on GitHub



The primary class website for this course is on GitHub



https://github.com/chaneyn/ESDA_CEE690-02

All lectures, assignments, and materials will be on the class' GitHub repository

The screenshot shows a GitHub repository page for 'chaneyn / ESDA_CEE690-02'. The repository name is 'Environmental Spatial Data Analysis'. Key statistics displayed include 42 commits, 2 branches, 0 releases, 1 contributor, and an MIT license. The commit history lists several recent changes:

File / Action	Description	Time
chaneyn Finalized syllabus	Finalized syllabus	1 minute ago
Assignments	Finalized syllabus	1 minute ago
Lectures	Added other info	6 days ago
Other	Added other info	5 days ago
LICENSE	Create LICENSE	3 days ago
README.md	Update README.md	6 minutes ago

A large text box at the bottom contains the repository's description: 'CEE690-02: Environmental Spatial Data Analysis'.

*Solutions to assignments will be posted on Sakai

Syllabus is on the class GitHub repository

README.md



CEE690-02: Environmental Spatial Data Analysis

Fall 2020

Course Information

Lectures are on Tuesdays and Thursdays from 3:30 PM - 4:45 PM. The course website is on GitHub (<https://github.com/chaneyn/CEE690-02>). Class announcements will be made via Sakai (CEE690.02.F20).

Instructor

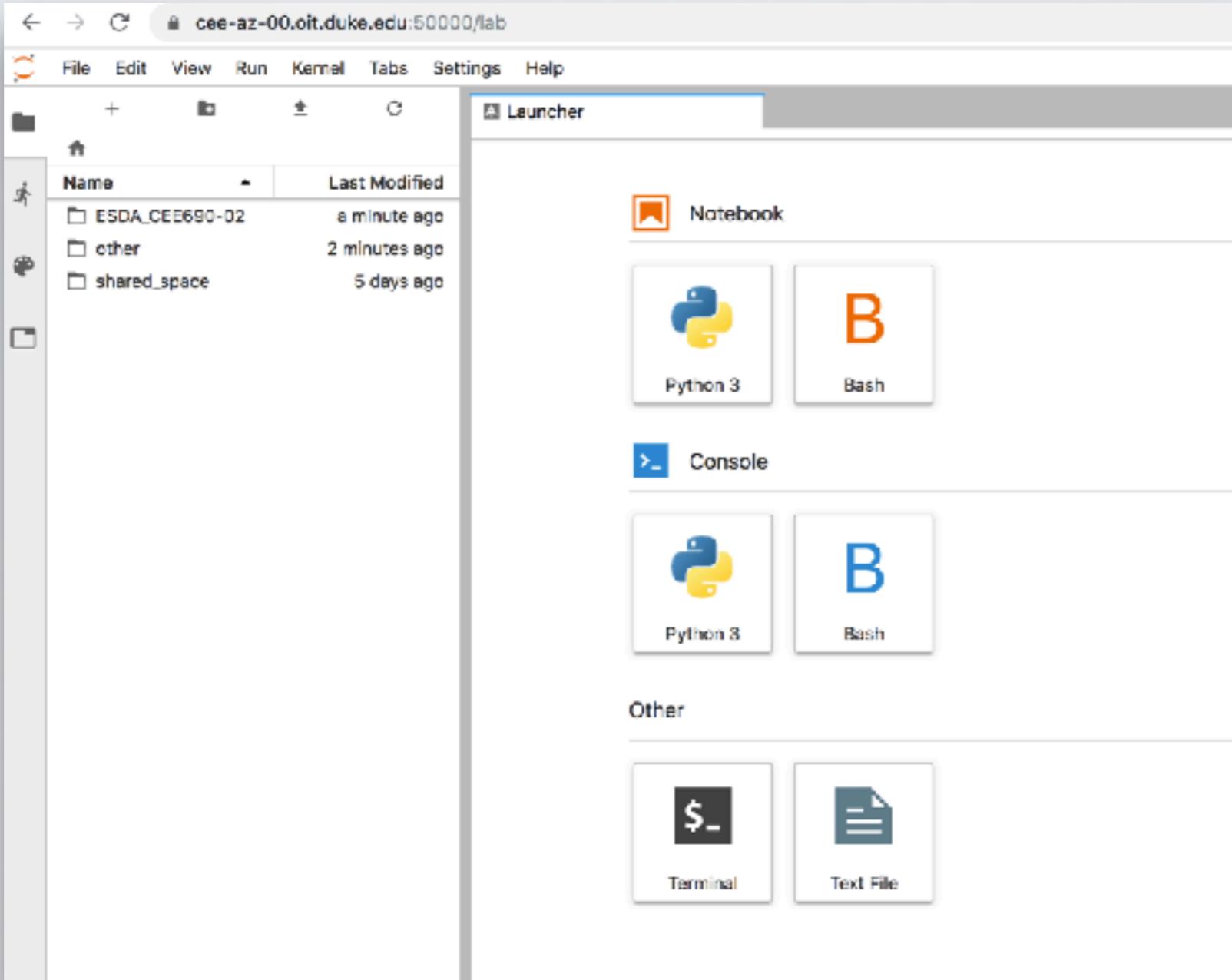
Professor Nathaniel W. Chaney (Nate)

Email: nathaniel.chaney@duke.edu

Office: FCIEMAS 2463

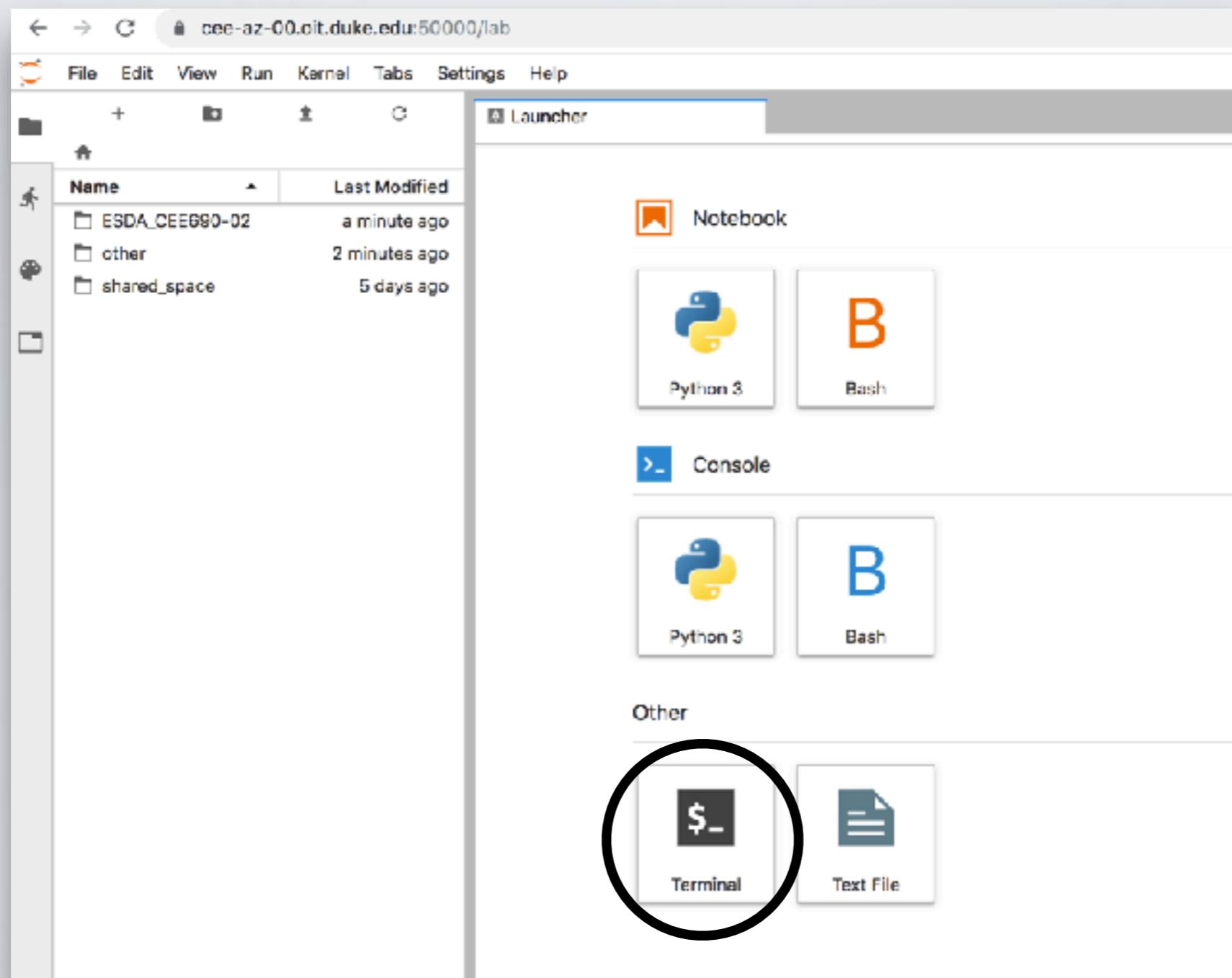
Office hours: Thursdays after class via Zoom

Getting started: Clone the course's GitHub repository



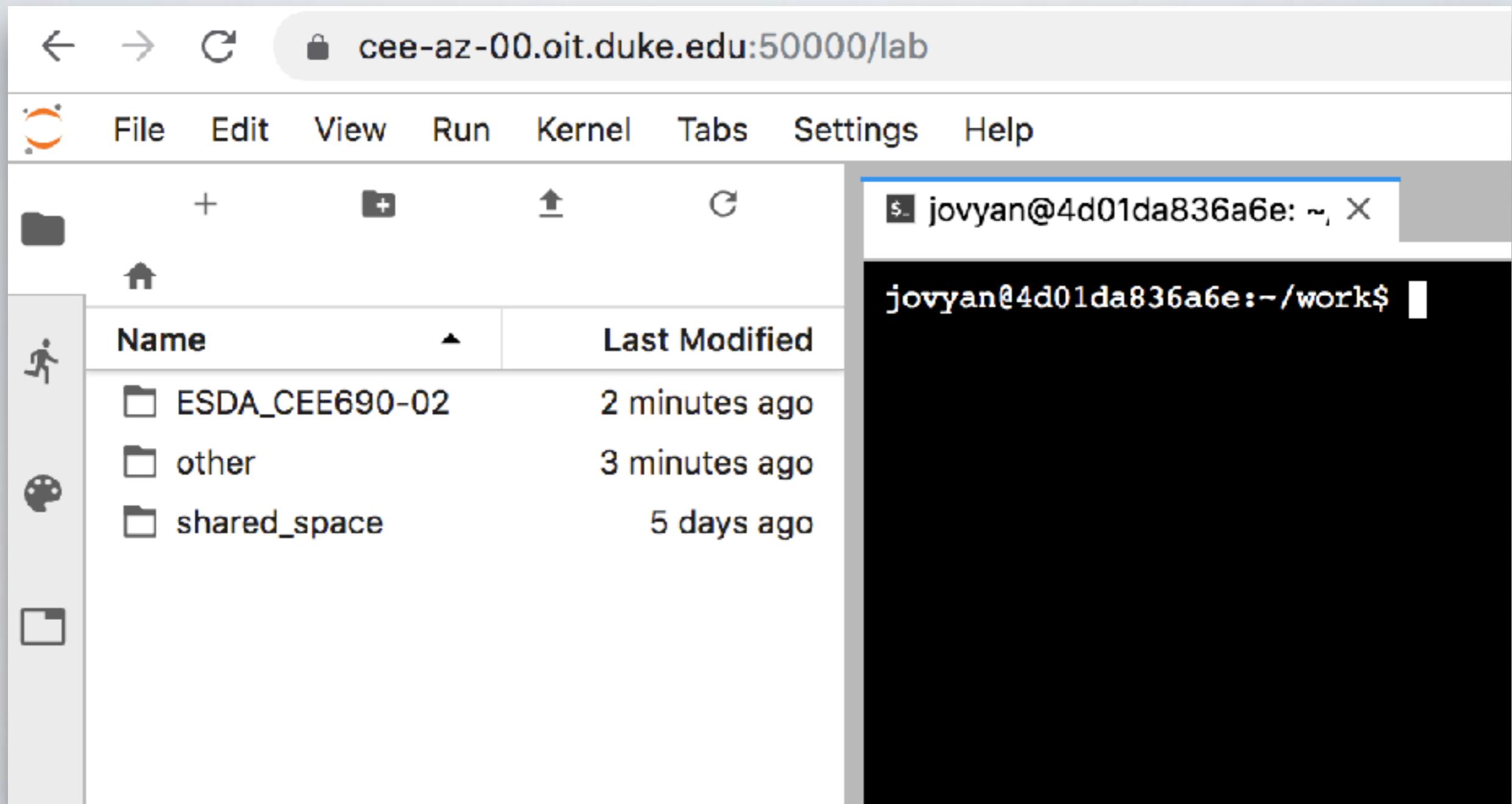
Navigate to your Docker container instance

Open a terminal !

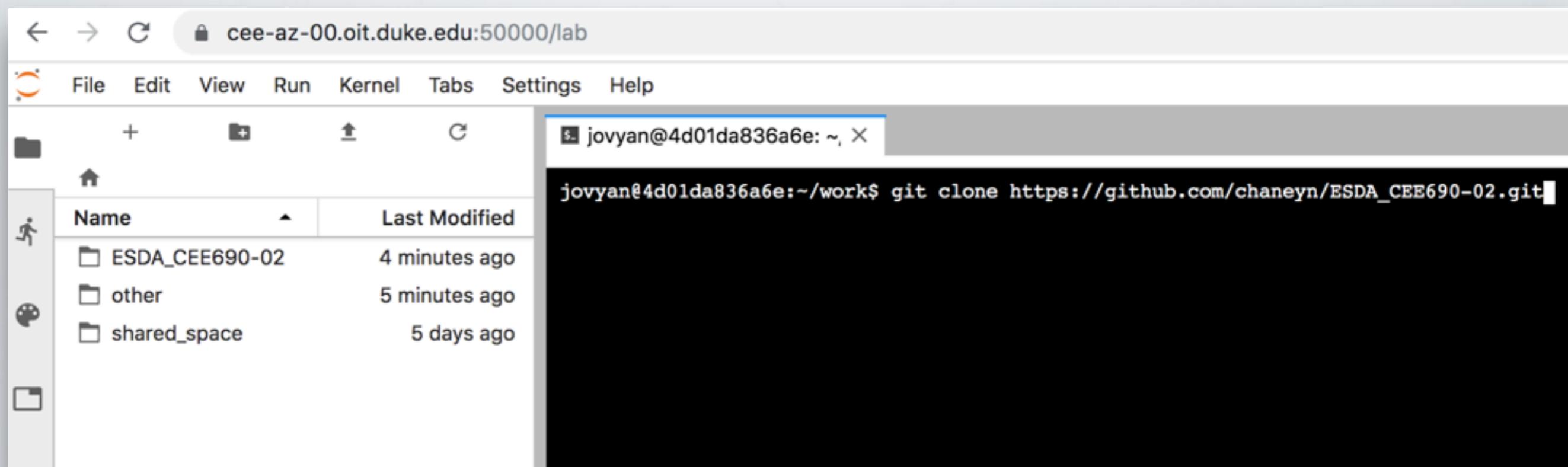


Press the terminal button

Open a terminal II



Write the following git command

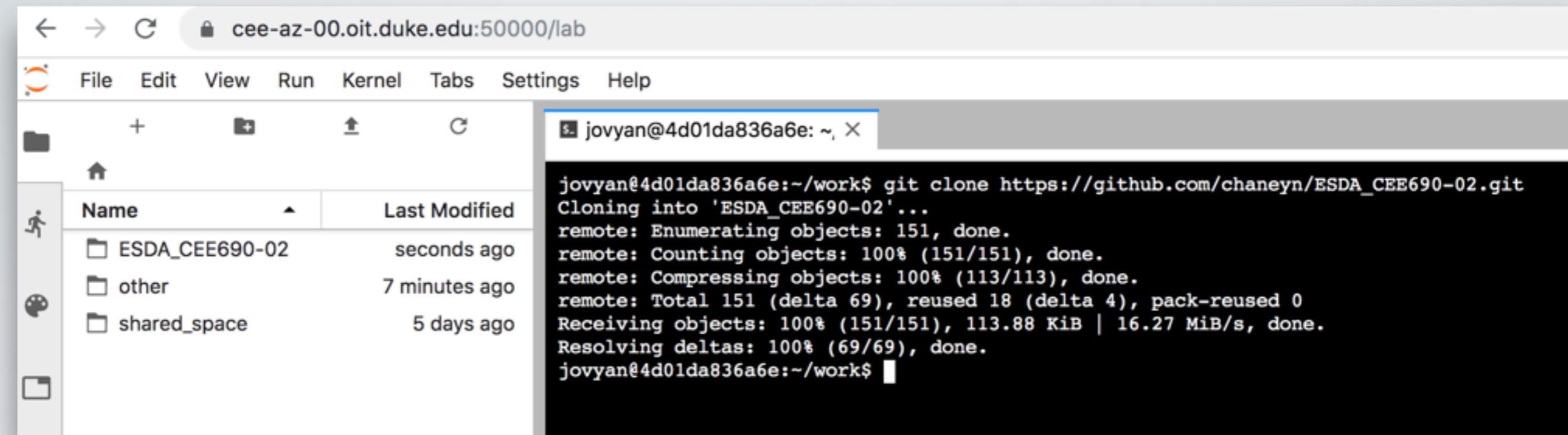


A screenshot of a terminal window titled "jovyan@4d01da836a6e: ~, X". The window shows a command being typed: "jovyan@4d01da836a6e:~/work\$ git clone https://github.com/chaneyn/ESDA_CEE690-02.git". The terminal is part of a desktop environment with a file manager sidebar on the left.

Name	Last Modified
ESDA_CEE690-02	4 minutes ago
other	5 minutes ago
shared_space	5 days ago

Then press enter

You will see something like this



The screenshot shows a Jupyter Notebook interface. The top bar includes navigation icons, a URL field with "cee-az-00.oit.duke.edu:50000/lab", and a menu bar with File, Edit, View, Run, Kernel, Tabs, Settings, and Help. On the left is a sidebar with icons for Home, User, Profile, and Help. Below the menu is a file browser with a list of files:

Name	Last Modified
ESDA_CEE690-02	seconds ago
other	7 minutes ago
shared_space	5 days ago

The main area contains a terminal window with the following output:

```
jovyan@4d01da836a6e:~/work$ git clone https://github.com/chaneyn/ESDA_CEE690-02.git
Cloning into 'ESDA_CEE690-02'...
remote: Enumerating objects: 151, done.
remote: Counting objects: 100% (151/151), done.
remote: Compressing objects: 100% (113/113), done.
remote: Total 151 (delta 69), reused 18 (delta 4), pack-reused 0
Receiving objects: 100% (151/151), 113.88 KiB | 16.27 MiB/s, done.
Resolving deltas: 100% (69/69), done.
jovyan@4d01da836a6e:~/work$
```

Congratulations you have cloned your first repository!

Explore the cloned directory

The screenshot shows a Jupyter Notebook interface. The top navigation bar includes back, forward, and refresh buttons, followed by the URL "cee-az-00.oit.duke.edu:50000/lab". Below the URL is a menu bar with File, Edit, View, Run, Kernel, Tabs, Settings, and Help. To the left is a sidebar with icons for user profile, course information, and file operations. The main area has two tabs: "jovyan@4d01da836a6e: ~" and "README.md". The "README.md" tab is active, displaying a Markdown file with course details.

```
# CEE690-02: Environmental Spatial Data Analysis
## Fall 2019

## Course Information
Lectures are on Tuesdays and Thursdays from 3:05 PM – 4:20 PM (https://github.com/chaneyn/CEE690-02). Class announcements will be posted here.

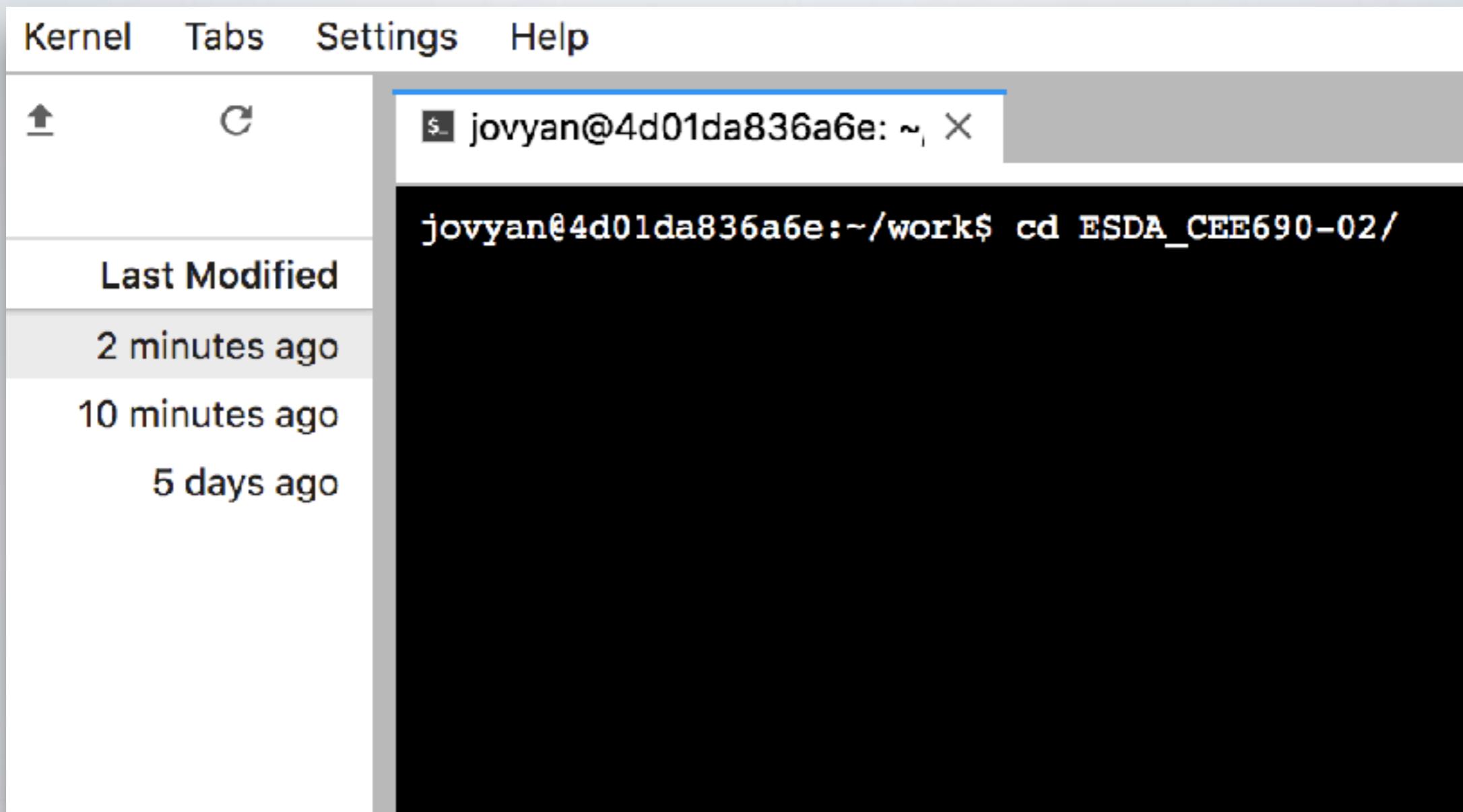
## Instructor
Professor Nathaniel W. Chaney (Nate)
Email: nathaniel.chaney@duke.edu
Office: FCIEMAS 2463
Office hours: By appointment on Friday mornings (calendly.co/nathanielchaney)

## TA
Reza Momenifar
Email: mohammadreza.momenifar@duke.edu
Office: FCIEMAS 2421
Office hours: Mondays 8:00–10:00 AM

## Course Description
This course provides an introduction on how to analyze and visualize spatial data using Python and R. The topics that will be covered include an overview of spatial data structures, conditional simulation, terrain analysis, dimensionality reduction, and machine learning. Python and R will be introduced and used to explore the listed topics.

## Prerequisites
```

Change directory at the terminal



A screenshot of a terminal window. The window has a menu bar with "Kernel", "Tabs", "Settings", and "Help". Below the menu is a toolbar with an "Up" arrow icon and a "C" icon. A sidebar on the left lists file or directory entries with "Last Modified" headers: "2 minutes ago", "10 minutes ago", and "5 days ago". The main terminal area shows a command line prompt: "\$ jovyen@4d01da836a6e: ~, X". Below the prompt, the command "jovyen@4d01da836a6e:~/work\$ cd ESDA_CEE690-02/" is entered. The terminal window has a dark background.

Then press enter

Git pull

The screenshot shows a terminal window with a light gray header bar containing a small circular icon and the text "jovyan@4d01da836a6e: ~, X". The main body of the terminal is black and contains white text. It shows a command-line session:

```
jovyan@4d01da836a6e:~/work$ cd ESDA_CEE690-02/
jovyan@4d01da836a6e:~/work/ESDA_CEE690-02$ git pull
Already up to date.
jovyan@4d01da836a6e:~/work/ESDA_CEE690-02$
```

To the left of the terminal window, there is a vertical sidebar with a light gray background. It has a header "C" and four items listed under "Last Modified":

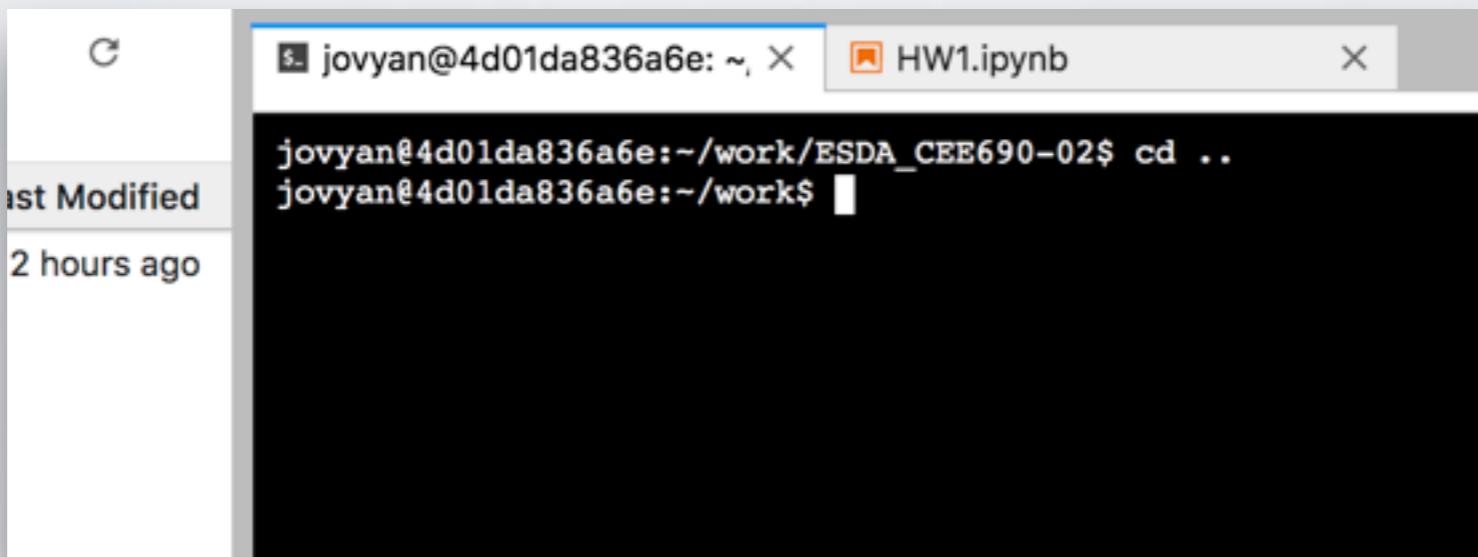
- 3 minutes ago
- 1 minutes ago
- 5 days ago

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)

Make a GitHub repository to turn in your assignments

Return to the previous directory



A screenshot of a terminal window titled "jovyan@4d01da836a6e: ~, X". The window shows the command "jovyan@4d01da836a6e:~/work/ESDA_CEE690-02\$ cd .." being typed. The terminal is running in a Jupyter Notebook environment, as indicated by the ".ipynb" icon in the title bar. The background of the slide features a blurred version of this terminal window.

**Create a directory (use your own
initials)**

```
jovyan@4d01da836a6e:~/work$ mkdir ESDA_nwc
```

Enter directory

```
jovyan@4d01da836a6e:~/work$ cd ESDA_nwc/
```

Create git repository

```
jovyan@4d01da836a6e:~/work/ESDA_nwc$ git init  
Initialized empty Git repository in /home/jovyan/work/ESDA_nwc/.git/  
jovyan@4d01da836a6e:~/work/ESDA_nwc$
```

Copy over assignment number 1

```
e:~/work/ESDA_nwc$ cp ../ESDA_CEE690-02/Assignments/HW1.ipynb .
```

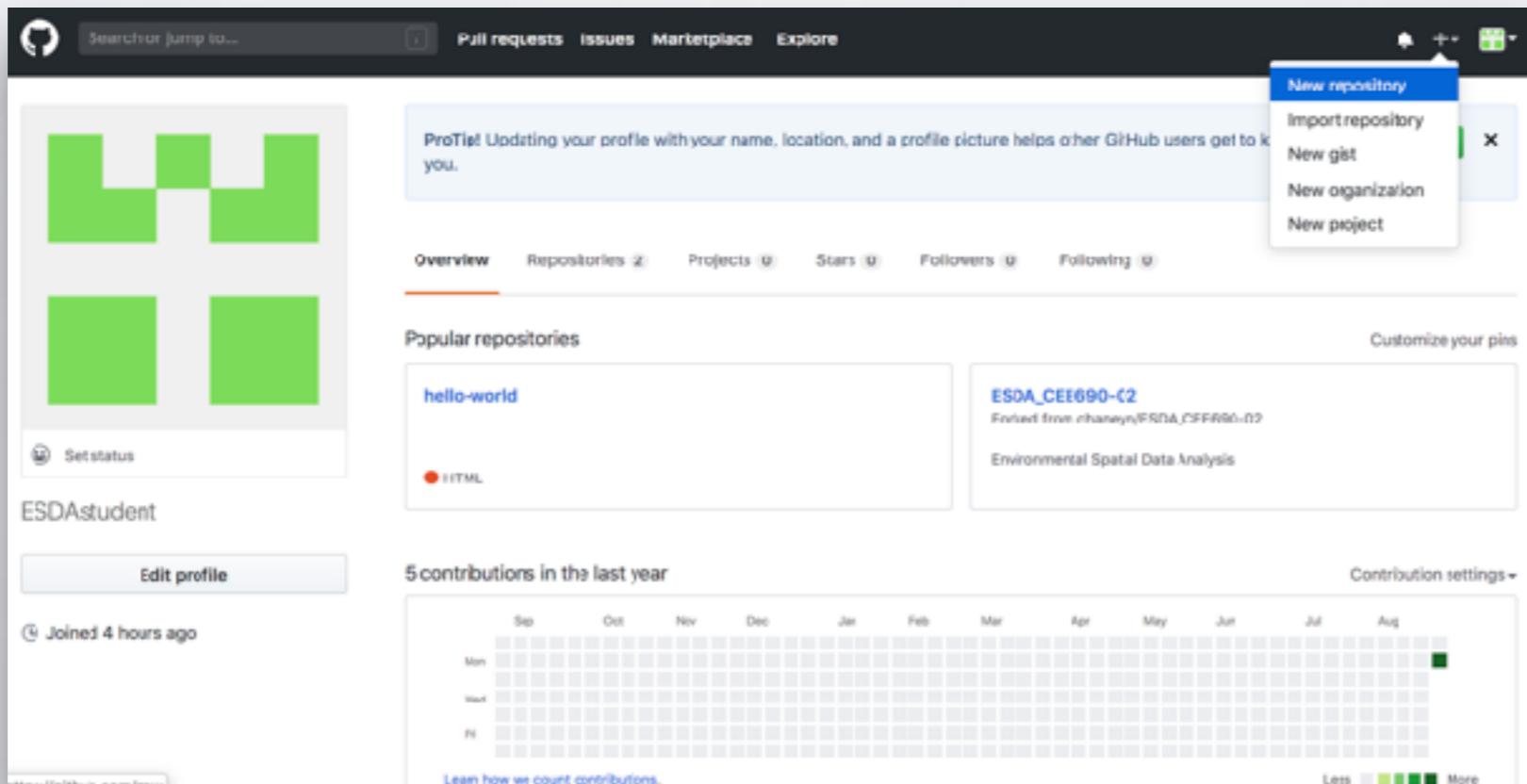
Add assignment to your repository

```
:~/work/ESDA_nwc$ git add HW1.ipynb
```

Commit changes to repository

```
jovyan@4d01da836a6e:~/work/ESDA_nwc$ git commit -am "Added HW1 file"
[master (root-commit) 7f144f0] Added HW1 file
 1 file changed, 34 insertions(+)
 create mode 100644 HW1.ipynb
```

Make a corresponding repository on your GitHub account



Naming your class GitHub repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner



ESDAstudent ▾

Repository name *

ESDA_nc153



Great repository names are short and memorable. Need inspiration? How about [reimagined-octo-guide](#)?

Make private (unless you want everyone to see your completed assignments!)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Create the repository!

Skip this step if you're importing an existing repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Attach both your local and online repositories (replace the link with yours)

...or push an existing repository from the command line

```
git remote add origin https://github.com/ESDAstudent/ESDA_nwc.git  
git push -u origin master
```



```
jovyan@4d01da836a6e:~/work/ESDA_nwc$ git push -u origin master  
Username for 'https://github.com': ESDAstudent  
Password for 'https://ESDAstudent@github.com':  
Counting objects: 3, done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 489 bytes | 489.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
To https://github.com/ESDAstudent/ESDA_nwc.git  
 * [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.  
jovyan@4d01da836a6e:~/work/ESDA_nwc$ █
```

Now they are attached to each other! And the online version has been updated.

The screenshot shows a GitHub repository page for the user 'ESDAstudent' with the repository name 'ESDA_nwc'. The repository is marked as 'Private'. At the top right, there are buttons for 'Unwatch' (with 1 watch), 'Star' (0 stars), and 'Fork' (0 forks). Below the header, there are navigation links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Security', 'Insights', and 'Settings'. A note below the links states 'No description, website, or topics provided.' with an 'Edit' button. There is also a 'Manage topics' link. Key statistics are displayed: 1 commit, 1 branch, and 0 releases. Below these stats, there are buttons for 'Branch: master ▾', 'New pull request', 'Create new file', 'Upload files', 'Find File', and a green 'Clone or download ▾' button. A commit history is shown, with the latest commit by 'chaneyn' adding 'HW1.ipynb' 8 minutes ago. A blue call-to-action bar at the bottom encourages users to 'Add a README with an overview of your project.' and features a green 'Add a README' button.

ESDAstudent / ESDA_nwc Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases

Branch: master ▾ New pull request Create new file Upload files Find File Clone or download ▾

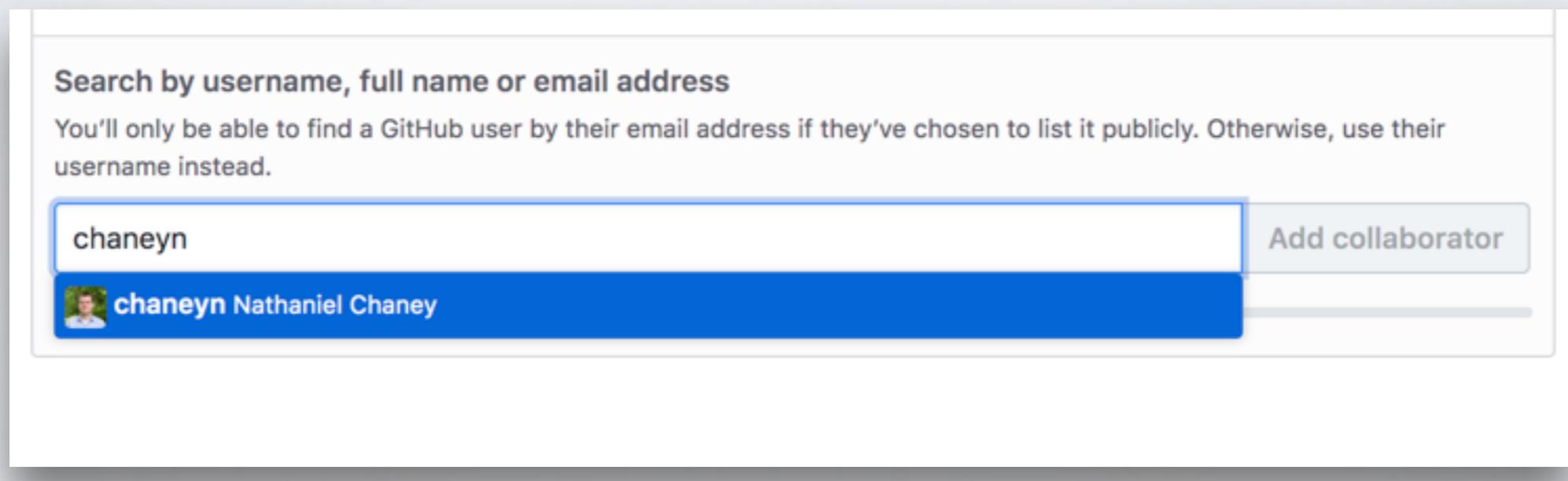
chaneyn Added HW1 file Latest commit 7f144f0 8 minutes ago

HW1.ipynb Added HW1 file 8 minutes ago

Add a README with an overview of your project. Add a README

Last thing you need to do is share your private repository with myself and the TA

On your repository go to “Settings” then “Collaborators”



With every new assignment:

- Copy the assignment from the class repository
- Add it to your private repository (git add)
- Commit the changes (git commit)
- Complete the assignment
- 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 nowadays to doing most research and industry data science. Forcing you to use version control throughout the course will ensure that you learn how to use it.

**Assignments submitted any other way will NOT
be accepted.**

Week 1 homework: Get your Docker container up and running, clone the class repository, create and link your private repository, and solve some introductory Python exercises.

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.