

CEE 690-02

ENVIRONMENTAL SPATIAL
DATA ANALYSIS (ESDA)

Lecture I

Introduction

- Instructor: Nathaniel Chaney (Nate)
- TA: Reza Momenifar

Motivation

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

Here are some examples from Hydrology

Floods

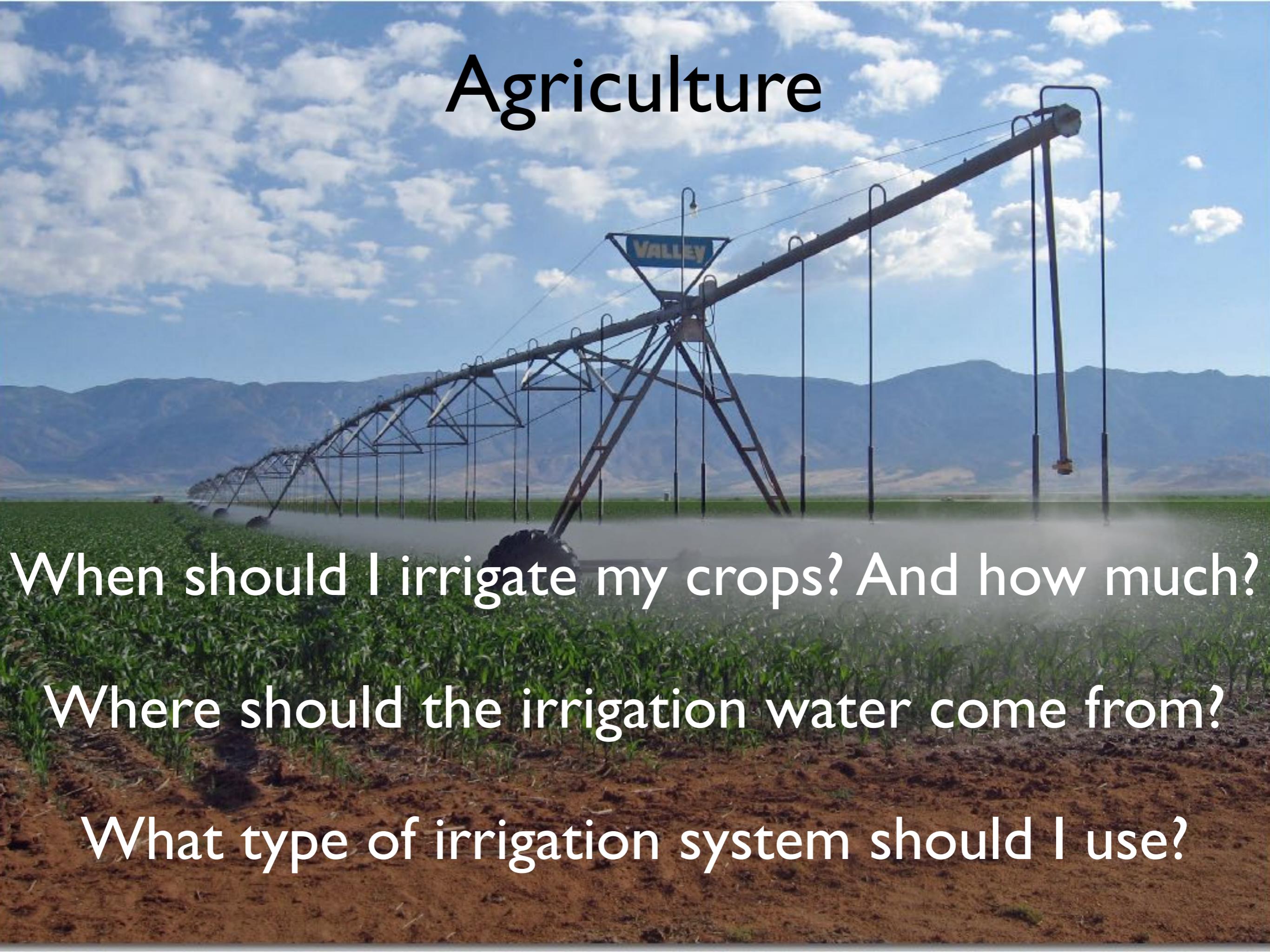
An aerial photograph of a residential street completely inundated by floodwater. The water covers the entire width of the street and reaches up to the windows of houses. Utility poles and wires are visible, partially submerged. A few houses are visible above the water level, and a red shipping container is visible in the background.

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

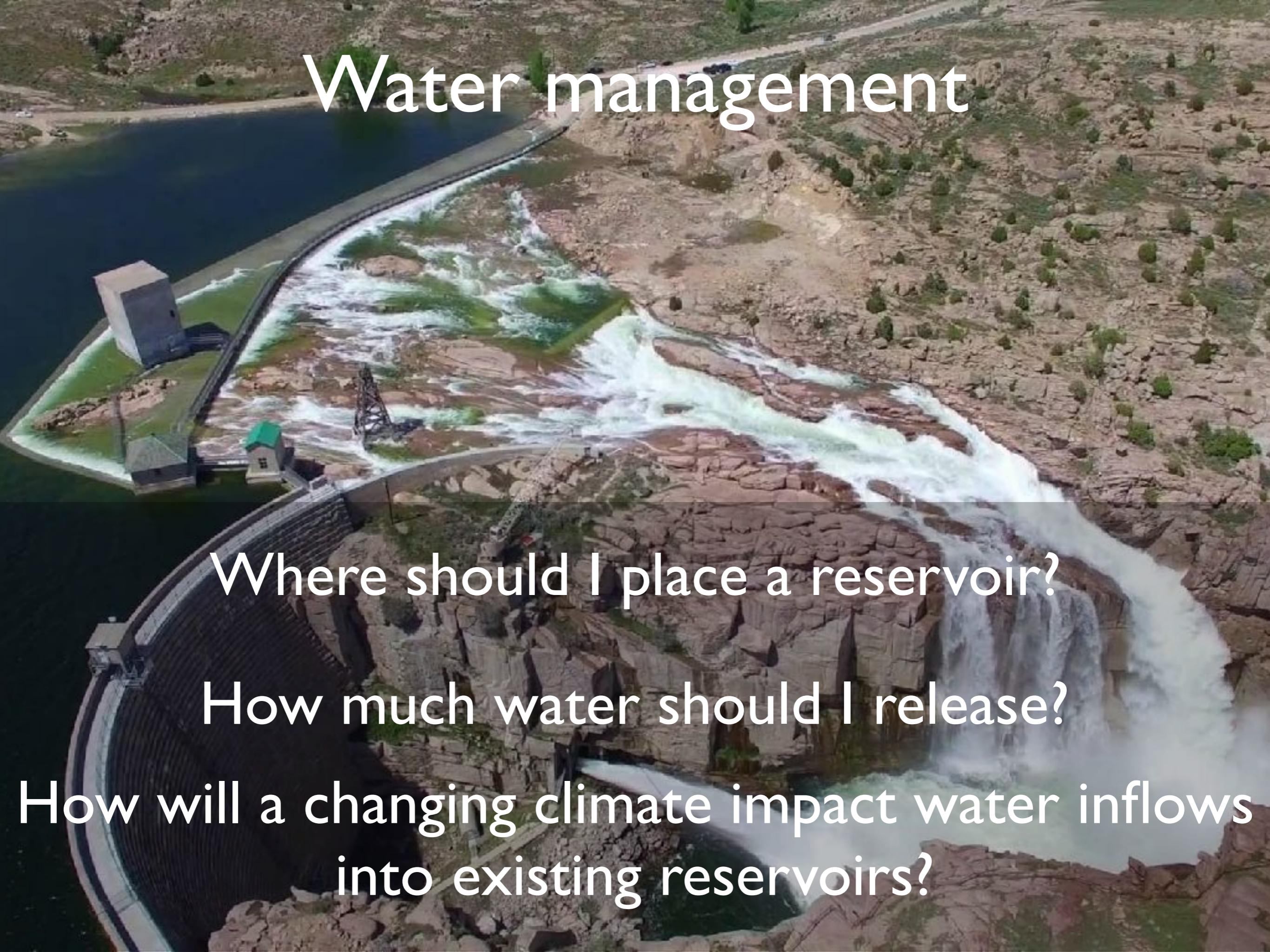


When should I irrigate my crops? And how much?

Where should the irrigation water come from?

What type of irrigation system should I use?

Water management

An aerial photograph of a large concrete dam with a curved top. Water flows over the dam's crest, creating white rapids. A long, narrow溢流道 (spillway) extends from the right side of the dam into a valley. The surrounding terrain is rocky and sparsely vegetated.

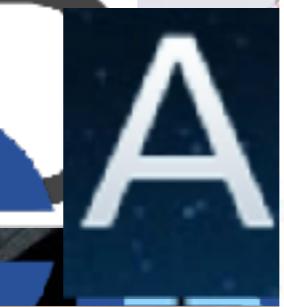
Where should I place a reservoir?

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



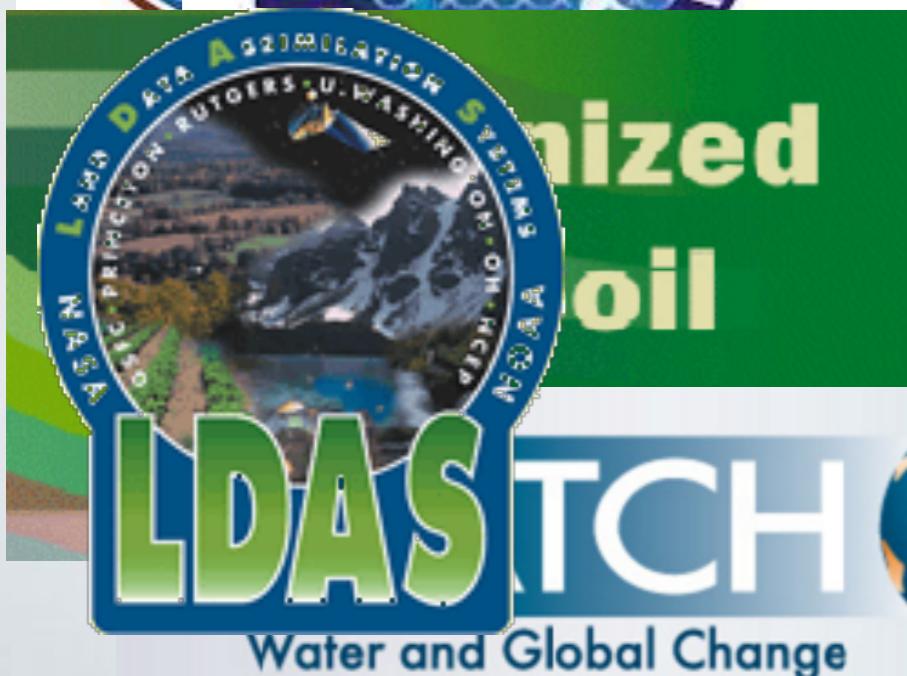
rest



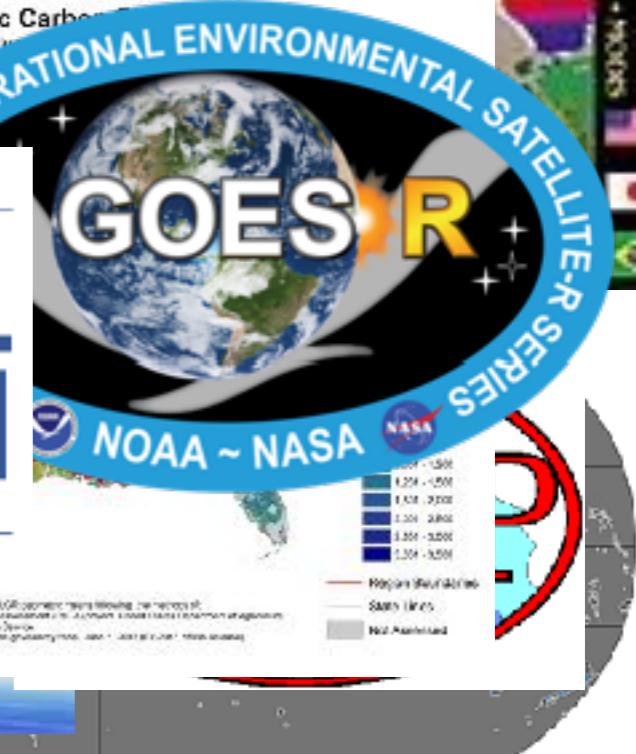
TANDEM monitor

A-40

“Age of Data”



THE LONG TERM
ECOLOGICAL
RESEARCH
NETWORK



Source: Proctor, J. et al. (2009) Geospatial data mining: the need for a
data quality audit. *Journal of Geographical Information Science*, 11, 1–12.
<http://dx.doi.org/10.1080/17412940802613700>

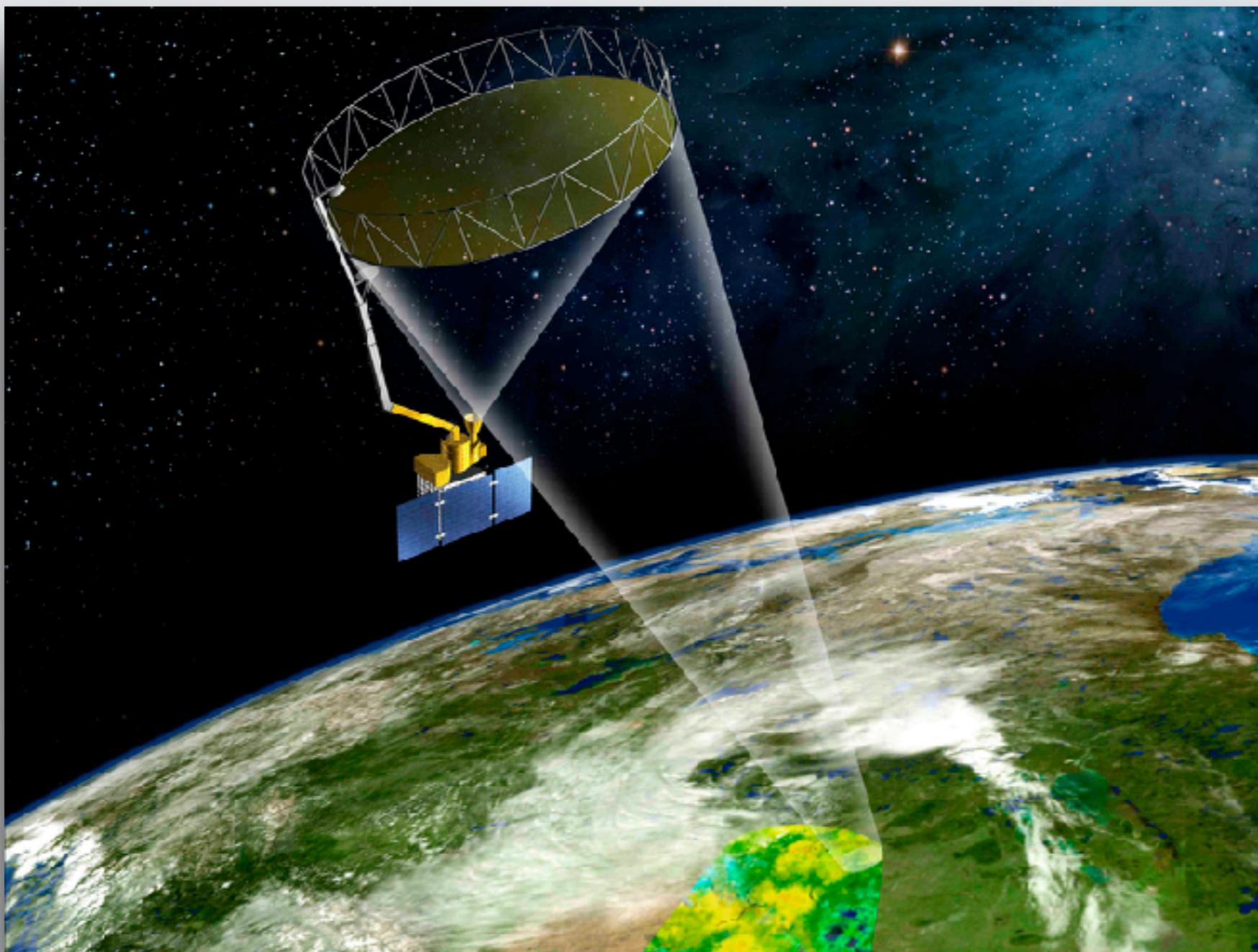
Report by
National Research Council
Released August 14, 2009

What Petabytes of data feels like

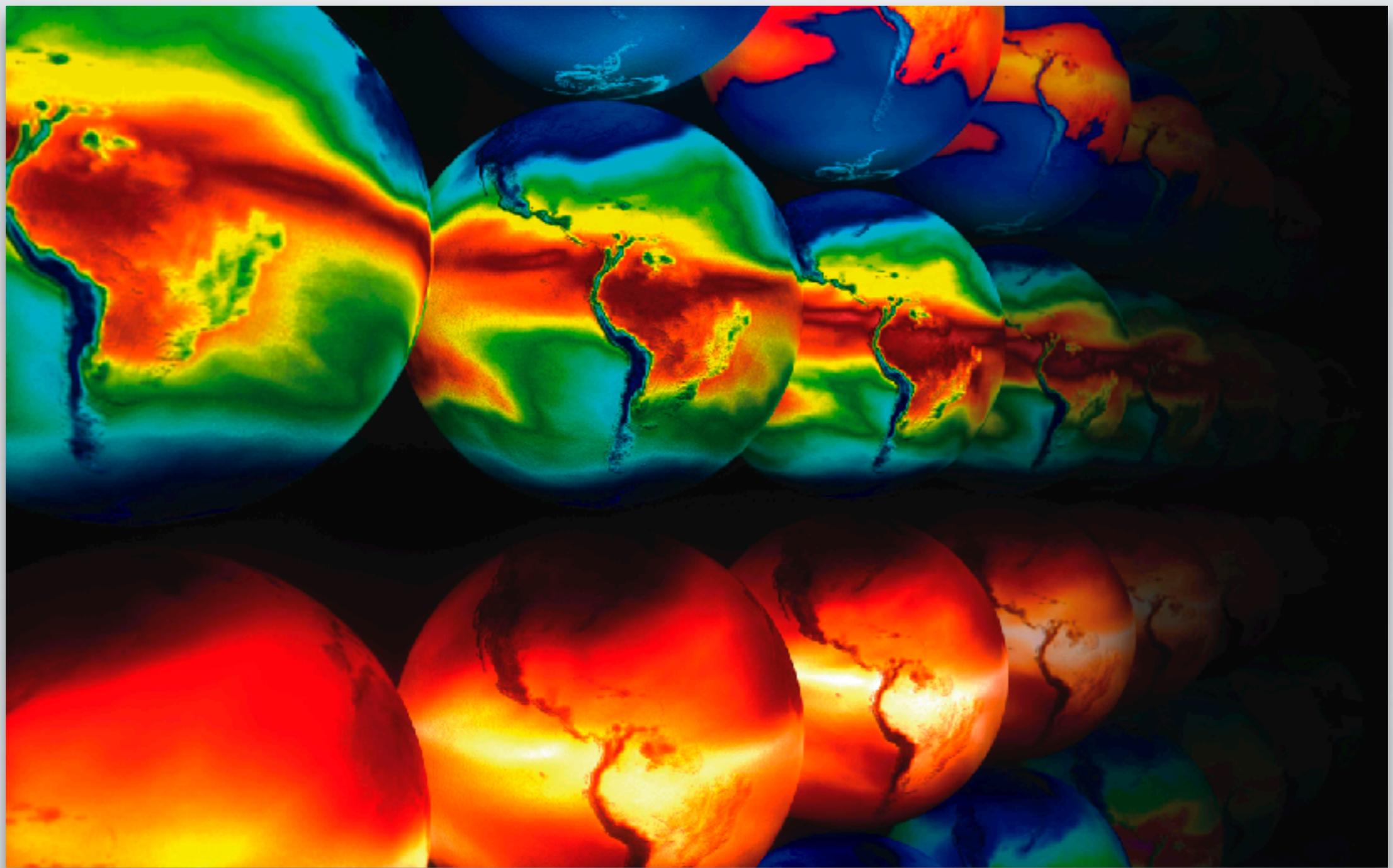


And the problem is even much
worse with spatial data

Example: Satellite remote sensing



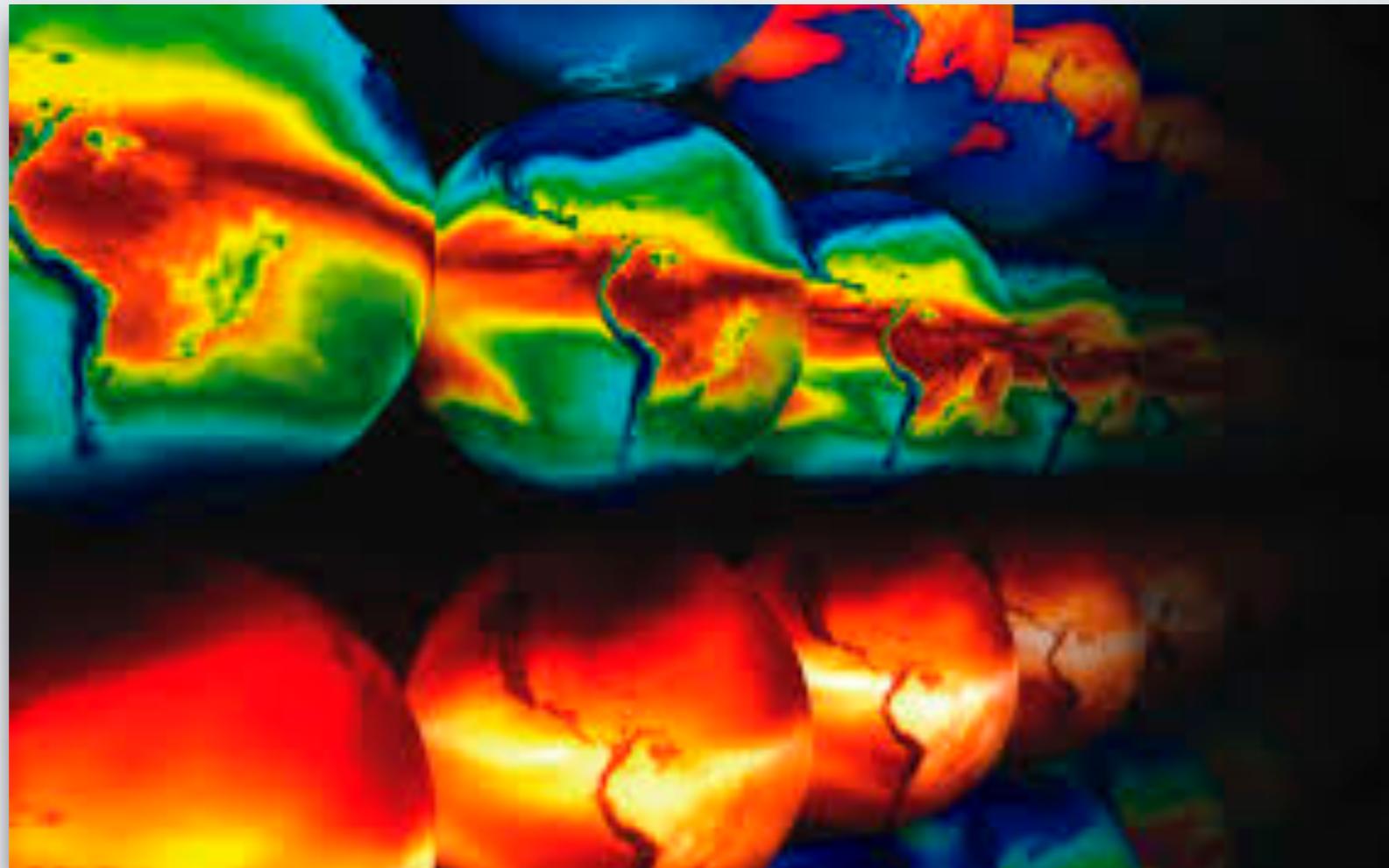
Example: Climate model output



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

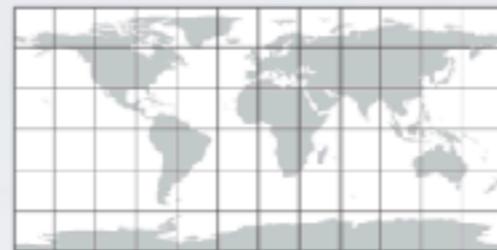


Plate Carrée



Winkel Tripel



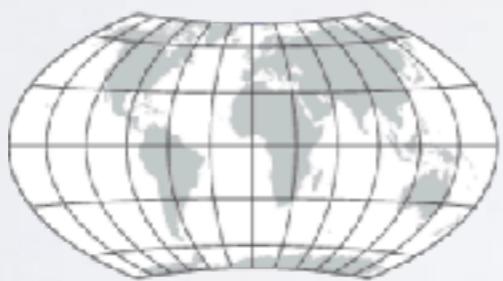
Eckert IV



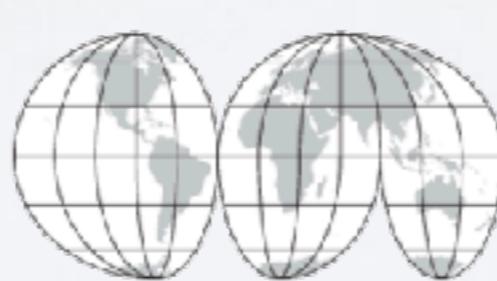
Mollweide



Mercator



Wagner VII



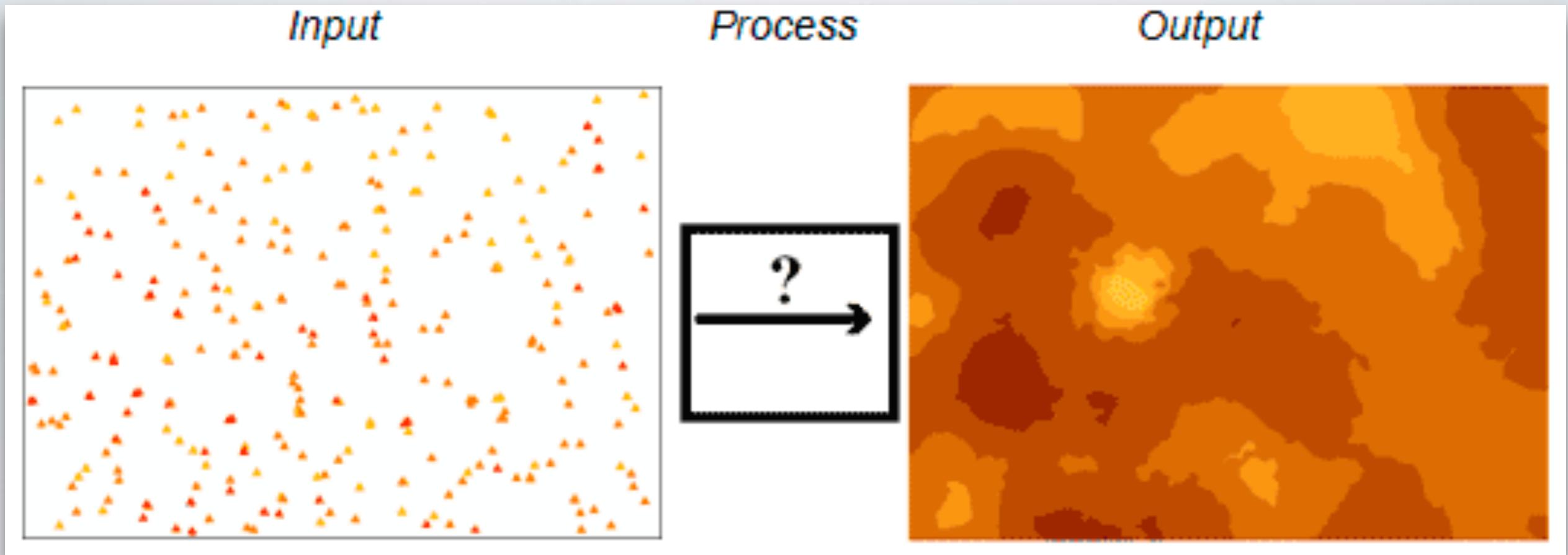
Interrupted Mollweide



Goode Homolosine

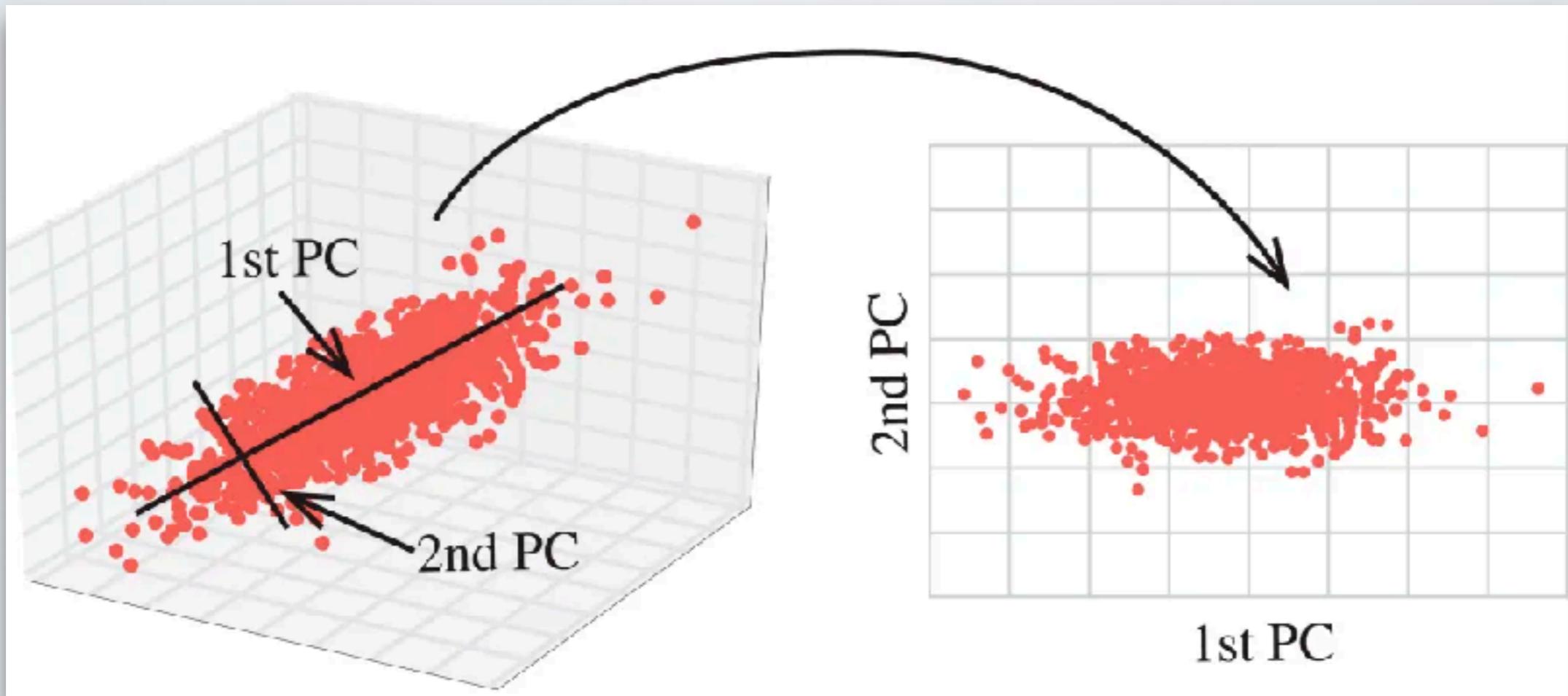
How can we transform a map to a different projection?

Topics: Spatial interpolation/ prediction



How do you create a spatial map from point data?

Topics: Reduce dimensions

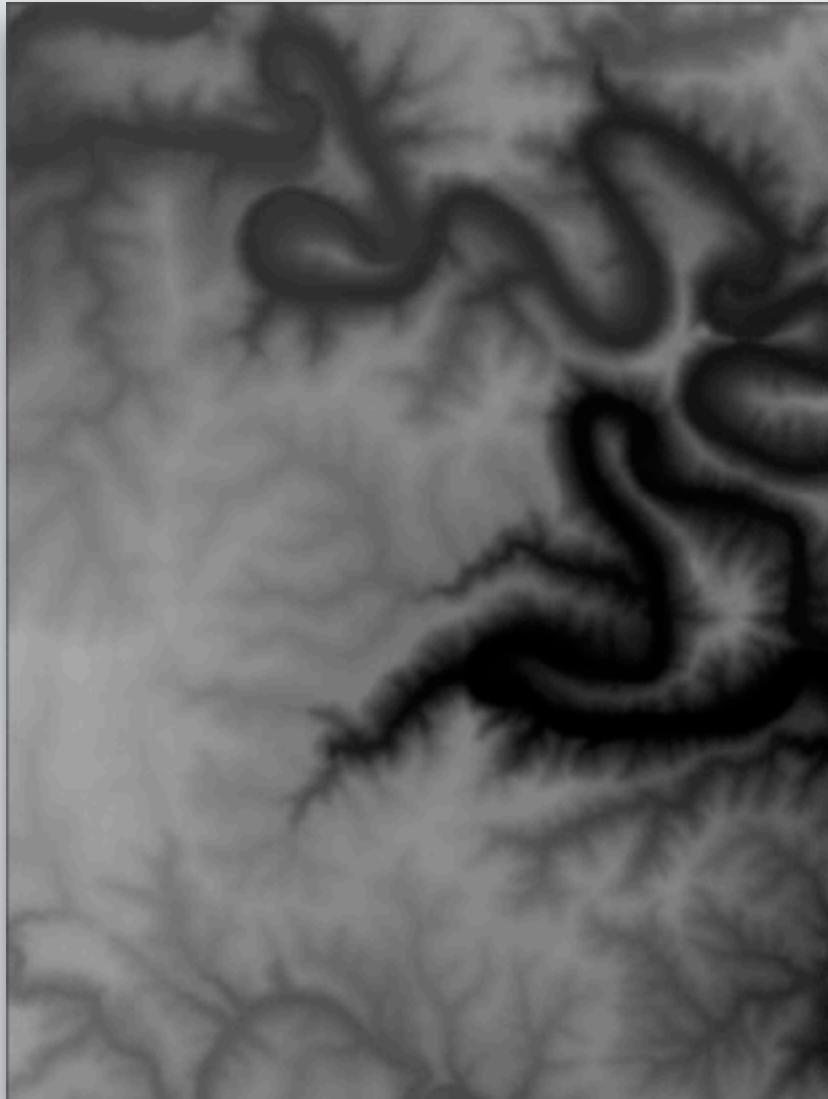


How do we compress data?

What are the most important features in the data?

Topics: Terrain analysis

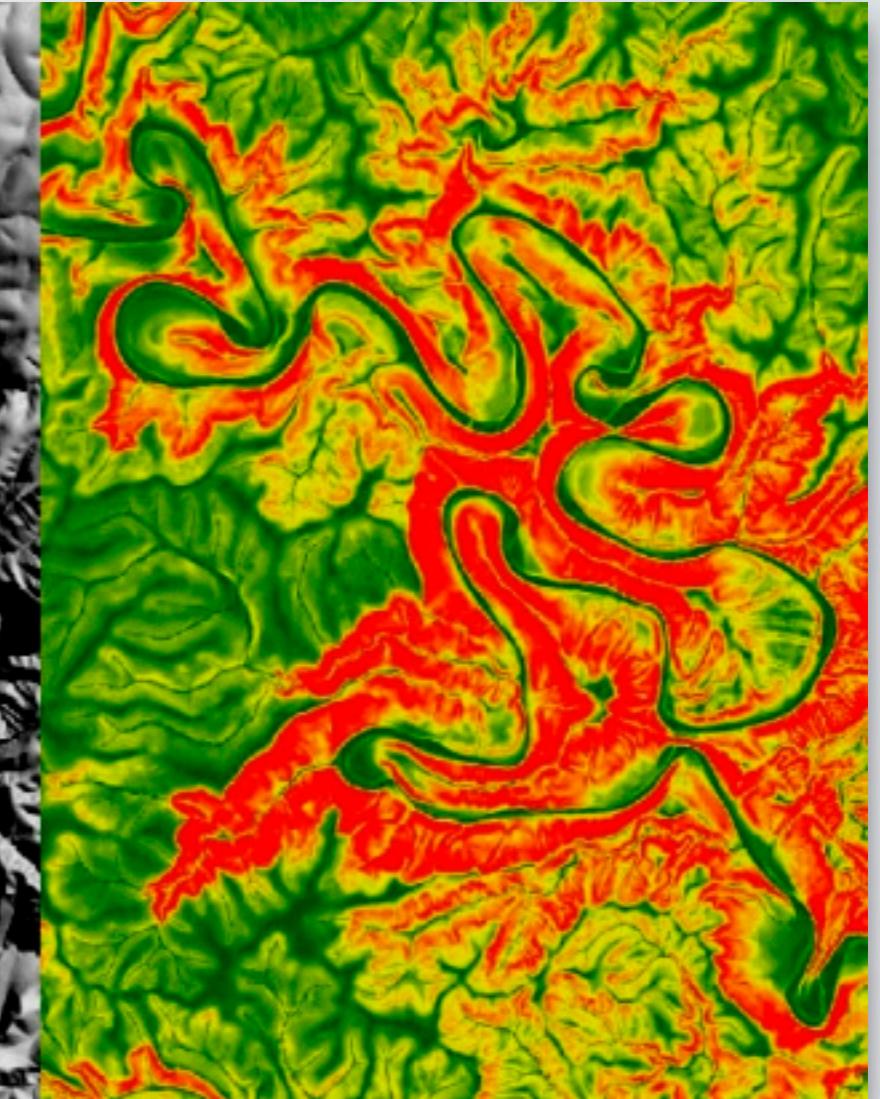
Elevation



Hillshading



Slope



What can we learn from elevation data?

And more...

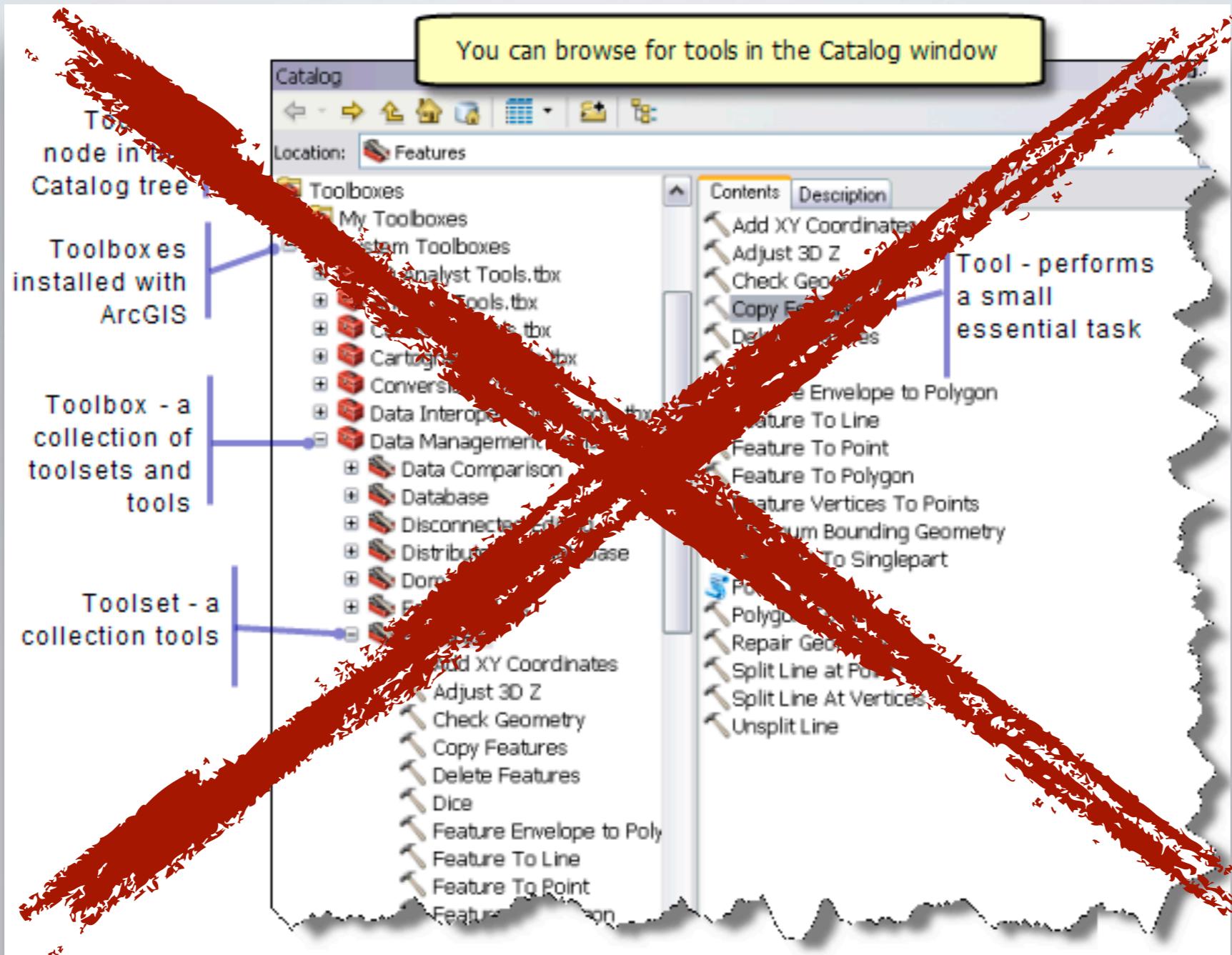
But what tools will we use to
learn how to create and analyze
these large datasets?

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: R



R level: Beginner

Secondary language: Bash



Bash level: Beginner*

Don't be afraid

We are going to spend a third of the class in learning how to program in Python (and some R 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: By appointment on Friday mornings (calendly.com/nathaniel-chaney)

TA

Reza Momenifar

Email: mohammadreza.momenifar@duke.edu

Office: FCIEMAS 2421

Office hours: Mondays 8:00-10:00 AM

Course Logistics: General Info

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

Textbooks?

There are none.

Welcome to the wild, wild west!

Homework

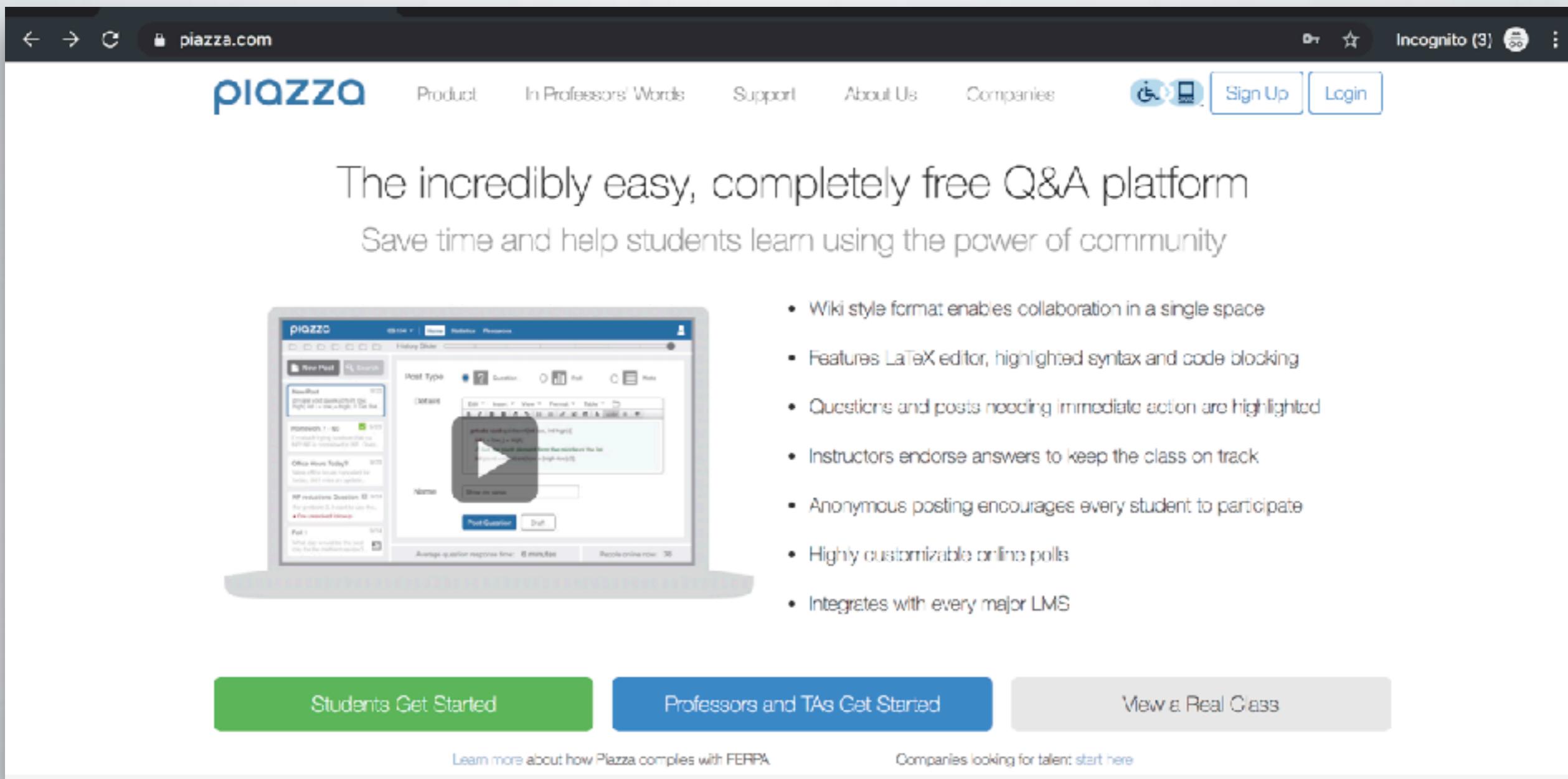
- Completed assignments will be submitted via each student's private GitHub repository
- Each assignment must be submitted before class on the day listed on the schedule.
- The homework with the lowest grade will be dropped
- Late homeworks will not be accepted
- Solutions to the homeworks will be provided via Sakai

Piazza: Working on homework

The screenshot shows the Piazza web interface for a class named "CEE 690-02". The top navigation bar includes links for "Q & A", "Resources", "Statistics", and "Manage Class". The user "Nathaniel Chaney" is logged in.

The main area displays a "New Post" form. The "Post Type" section has three options: "Question" (radio button), "Note" (selected radio button), and "Poll/In-Class Response". Below this, the "Post to" section shows "Entire Class" selected. The "Select Folder(s)" section lists several categories: hw1 (selected), hw2, hw3, hw4, hw5, hw6, hw7, hw8, hw9, hw10, project, exam, logistics, and other. A note states: "Your class is currently set to use default folders. Edit these folders via Manage Class page." The "Summary" section allows for a one-line summary (100 characters or less) with a placeholder "Enter a one line summary...". The "Details" section uses a plain text editor. A rich text editor toolbar is visible above the main text area, featuring buttons for Edit, Insert, View, Format, Table, Bold, Italic, Underline, etc. The bottom of the screen shows performance metrics: Average Response Time (N/A), Special Mentions (There are no special mentions at this time), Online Now (1), and This Week (1).

Piazza: Join the CEE 690-02 group (Environmental Spatial Data Analysis)



The screenshot shows the Piazza homepage with the URL piazza.com in the address bar. The page features the Piazza logo and navigation links for Product, In Professors' Words, Support, About Us, Companies, Sign Up, and Login. The main headline reads "The incredibly easy, completely free Q&A platform" followed by the subtext "Save time and help students learn using the power of community". A large image of a computer monitor displays the Piazza interface, which includes a sidebar with course announcements and a central area for posting questions and answers. To the right of the interface, a bulleted list highlights features: Wiki style format, LaTeX editor, highlighted syntax and code coloring, immediate action highlighting, instructor endorsements, anonymous posting, highly customizable online polls, and integration with major LMSs. At the bottom, three calls-to-action buttons are visible: "Students Get Started" (green), "Professors and TAs Get Started" (blue), and "View a Real Class" (grey).

- Wiki style format enables collaboration in a single space
- Features LaTeX editor, highlighted syntax and code coloring
- Questions and posts needing immediate action are highlighted
- Instructors endorse answers to keep the class on track
- Anonymous posting encourages every student to participate
- Highly customizable online polls
- Integrates with every major LMS

[Students Get Started](#) [Professors and TAs Get Started](#) [View a Real Class](#)

Learn more about how Piazza complies with FERPA. Companies looking for talent: start here.

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



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



Participation II:

Assignment review

- Beginning of next lecture after homework is due: Presentation to review the solutions from the previous homework
- Groups of students will be selected to lead these presentations (we will do the first one next week)
- Number of presenters will depend on the length and complexity of the homework

Final Project

The final project will be done in groups of 3-4 students and will involve the following components:

- Initial Proposal (October 10th via email)
 - 1 page, Single-spaced, 12 point font size, 1 inch margin
 - Contains: Title, motivation, and sketch of proposed methodology
- Proposal presentations (November 5th)
 - 10 minute informal oral presentation
 - Open discussion/feedback in class
- Revised Proposal (November 10th via email)
 - 3 pages, Single-spaced, 12 point font size, 1 inch margin
 - Contains: Title, introduction, objectives, Methodology, and a timeline of tasks
- Oral presentation (December 3rd and 5th in class)
 - 20 minute oral presentation, 5 minutes for questions
 - Everyone needs to be present for each presentation

Tentative schedule I

Date	Topic	New Software	Assignments	Presentation
08/27	Introduction	Jupyter/GitHub/Bash		
08/29	Python overview	Python		
09/03	Multi-dimensional arrays I	NumPy	HW #1 due	
09/05	Visualizing data I	Matplotlib/Seaborn		Reza and Nate
09/10	Raster data	Rasterio	HW #2 due	
09/12	Data storage	Pickle/H5py/NetCDF/Tiff		TBD
09/17	Multi-dimensional arrays II	Pandas/Xarray	HW #3 due	
09/19	Map projections I	GDAL		TBD
09/24	Map projections II		HW #4 due	
09/26	Visualizing data II	Cartopy		TBD

Tentative schedule II

10/01	Spatial statistics I	Scipy	HW #5 due	
10/03	Spatial statistics II			TBD
10/08	Fall Break			
10/10	R overview	R/Rpy2	Initial proposal due	
10/15	Spatial statistics III	Gstat (R)	HW #6 due	
10/17	Semivariogram			TBD
10/22	Kriging			
10/24	Conditional Simulation			
10/29	Spatial prediction I	Scikit-learn		
10/31	Spatial prediction II			

Tentative schedule III

11/05	Proposal presentations			
11/07	Dimensionality Reduction I		Revised proposal due	
11/12	Dimensionality Reduction II		HW #7 due	
11/14	Terrain Analysis I	TBD		TBD
11/19	Terrain Analysis II			
11/21	Vector data	Shapely		
11/26	Speed up code	Numba/Mpi4py/Dask		
11/28	Thanksgiving Break			
12/03	Oral Presentations			
12/05	Oral Presentations			
12/16	Written report due		HW #8 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" which says: "VM-Manage is a reservation service for compute environments such as RStudio, Eclipse, Jupyter, Spyder+Anaconda Python and other course-oriented Docker containers." Below this is a callout box stating: "For virtual machine reservations visit the new **Virtual Computing Manager** service at: <https://vcm.duke.edu>". On the right, there's a section titled "Docker containers" which says: "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." Below this is another callout box with the text: "Click here to access your personal Docker containers." At the bottom of the page is a footer with the Duke University logo, copyright information ("© 2012-2016 Duke University, Durham, NC 27708 | (919) 684-8111"), and links for "Accessibility", "Style Guide", "About Duke.edu", "Contact Us", and social media icons for YouTube, Twitter, Facebook, and RSS.

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 | (919) 684-8111 | 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](#)

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

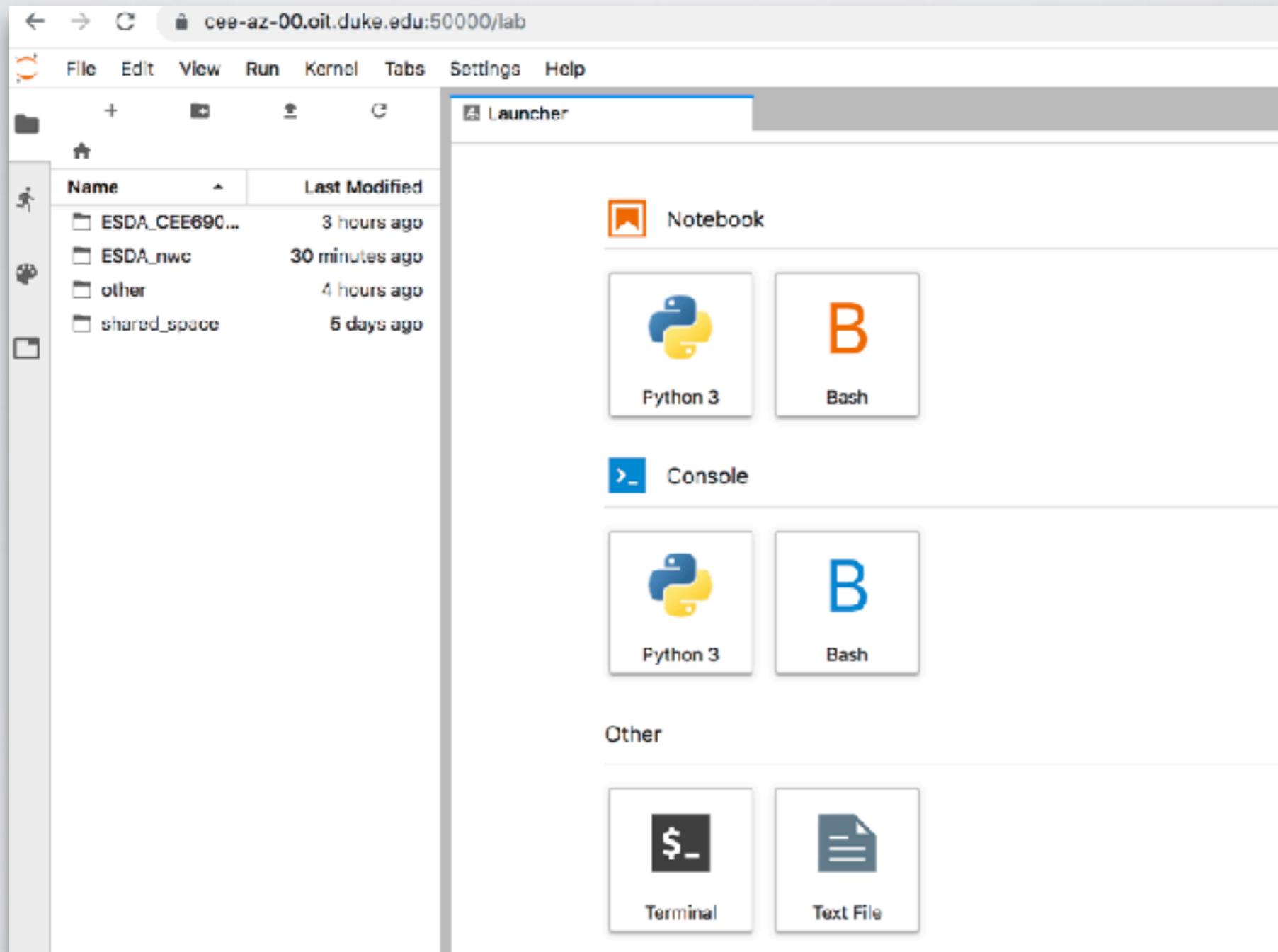
If you are unable to connect to your CEE690 environment you can [request a restart](#).

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

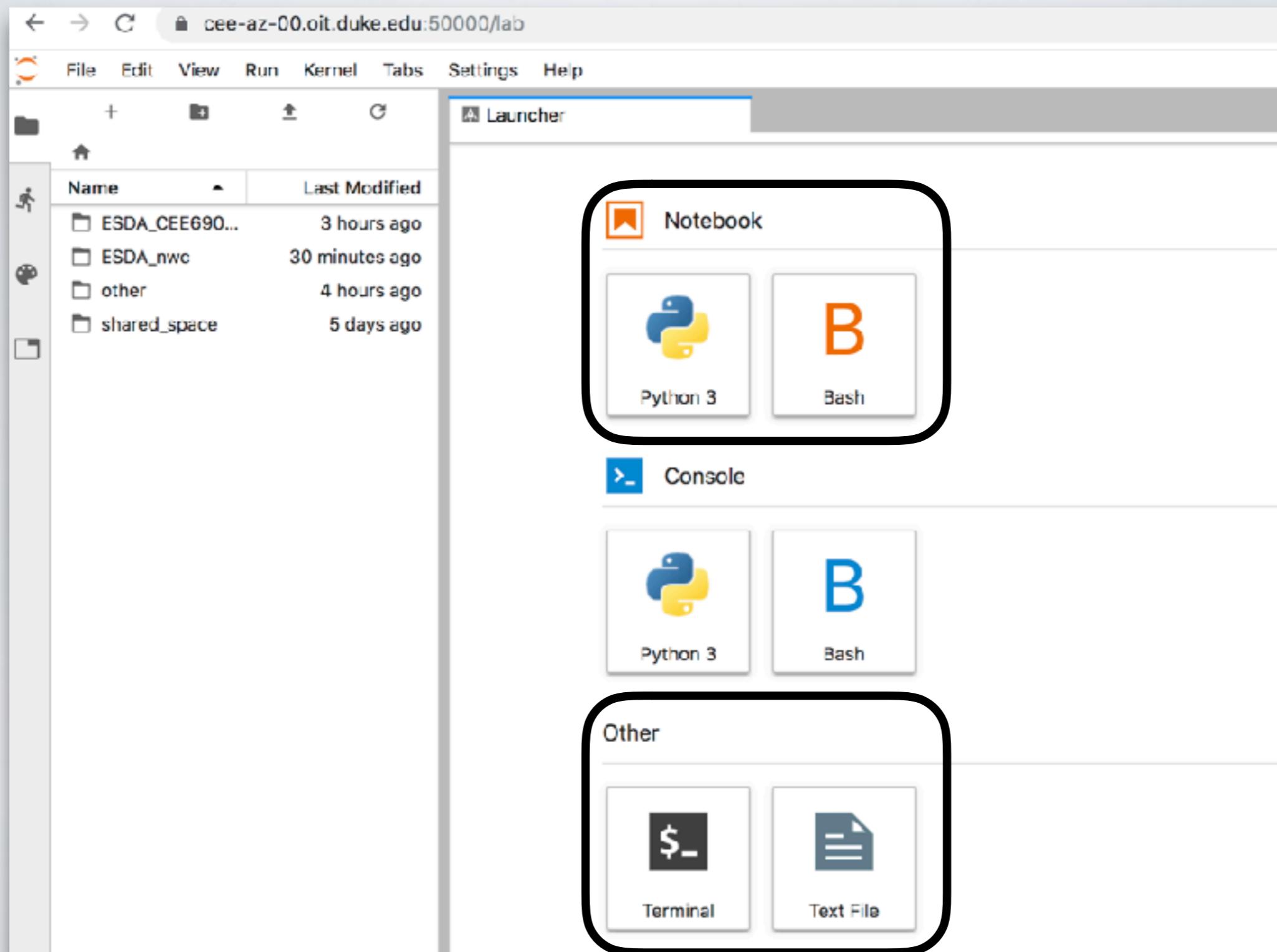


<https://jupyterlab.readthedocs.io/en/stable/>

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 is currently being edited, indicated by a blue selection bar.

```
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



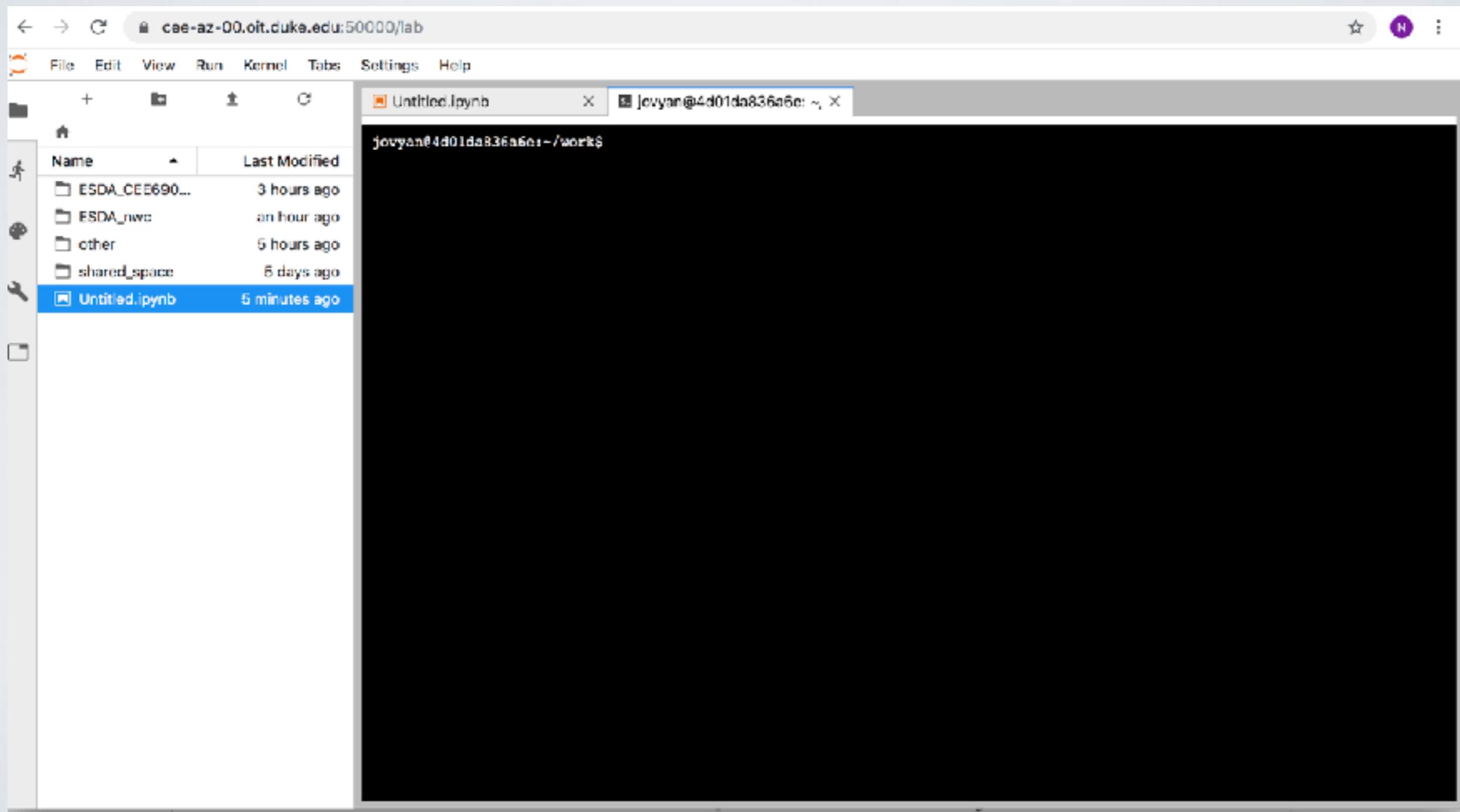
Remains one of the best ways to interact more directly with the computer (flexibility!)

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



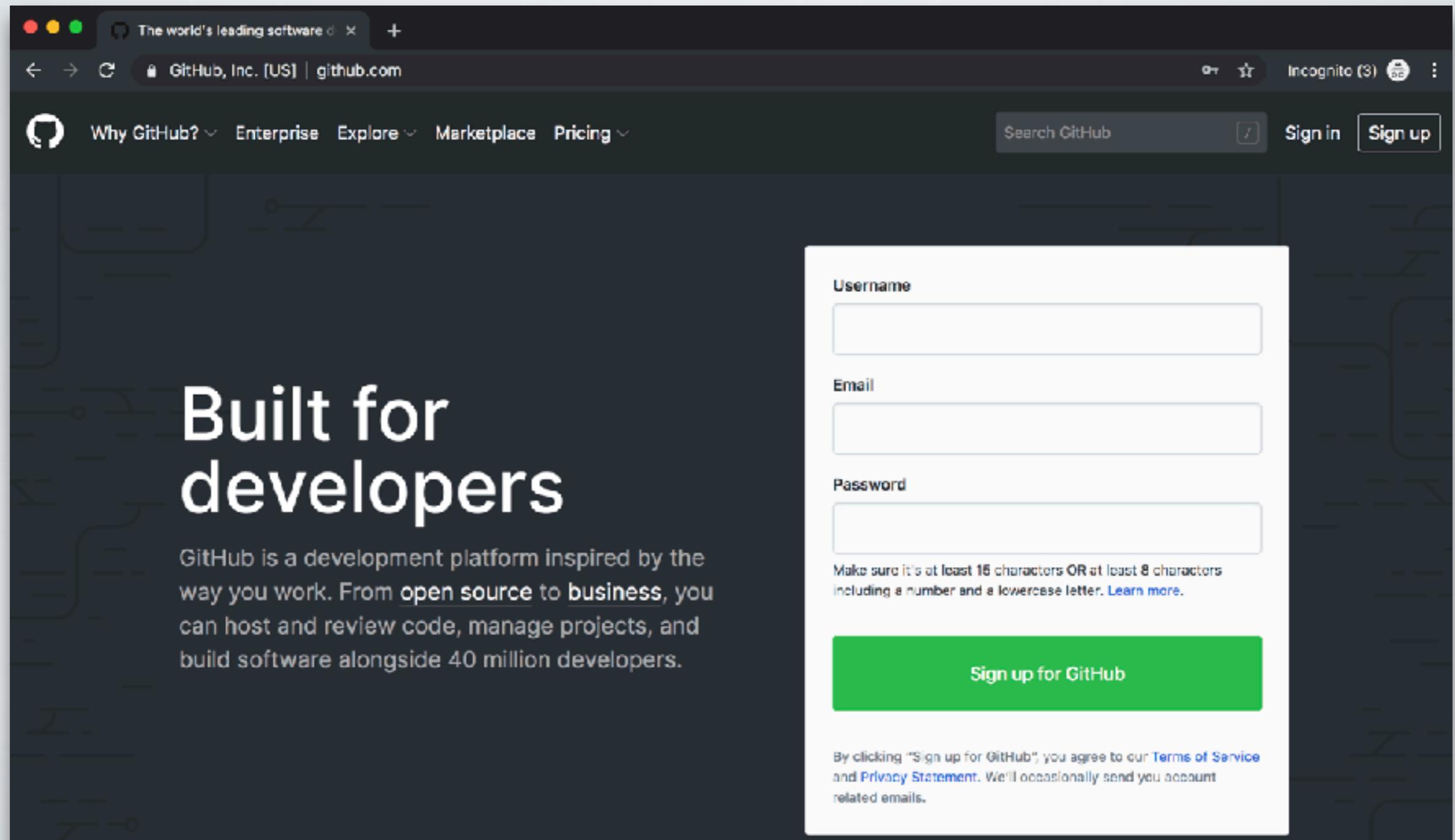
[https://www.youtube.com/watch?](https://www.youtube.com/watch?v=w3jLJU7DT5E)
[v=w3jLJU7DT5E](https://www.youtube.com/watch?v=w3jLJU7DT5E)

Version control: Git

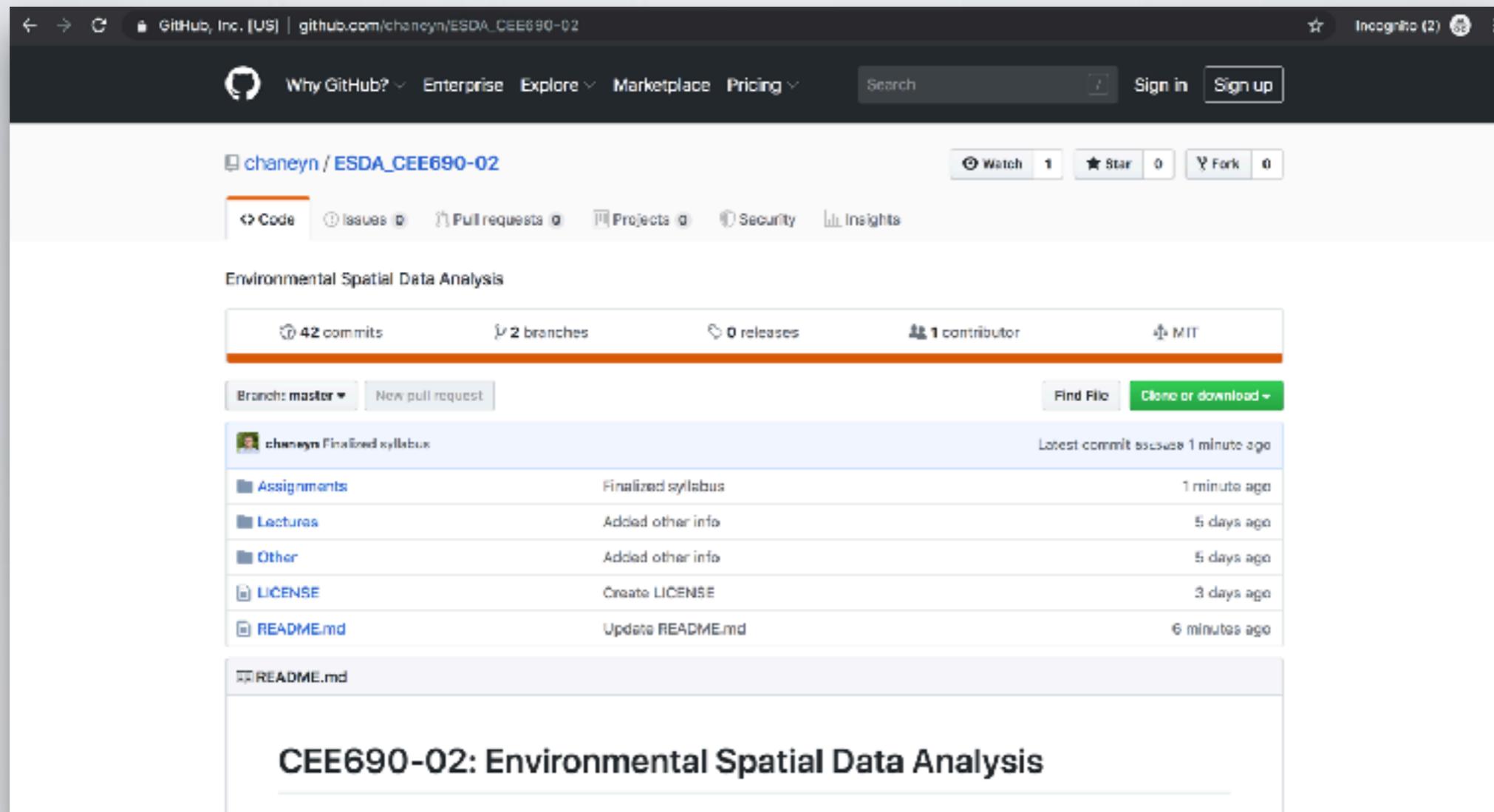


<https://vimeo.com/41027679>

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 files: 'Finalized syllabus', 'Assignments', 'Lectures', 'Other', 'LICENSE', and 'README.md'. The most recent commit was made 1 minute ago. A large banner at the bottom of the repository page reads 'CEE690-02: Environmental Spatial Data Analysis'.

File	Commit Message	Time Ago
Finalized syllabus	Finalized syllabus	1 minute ago
Assignments	Added other info	6 days ago
Lectures	Added other info	5 days ago
Other	Create LICENSE	3 days ago
LICENSE	Update README.md	6 minutes ago
README.md		

*Solutions to assignments will be posted on Sakai

The class syllabus is on the GitHub repository

A screenshot of a GitHub repository page for 'CEE690-02: Environmental Spatial Data Analysis'. The repository has three recent commits:

- Added other info (5 days ago)
- Create LICENSE (3 days ago)
- Update README.md (7 minutes ago)

The README.md file is open, displaying the following content:

CEE690-02: Environmental Spatial Data Analysis

Fall 2019

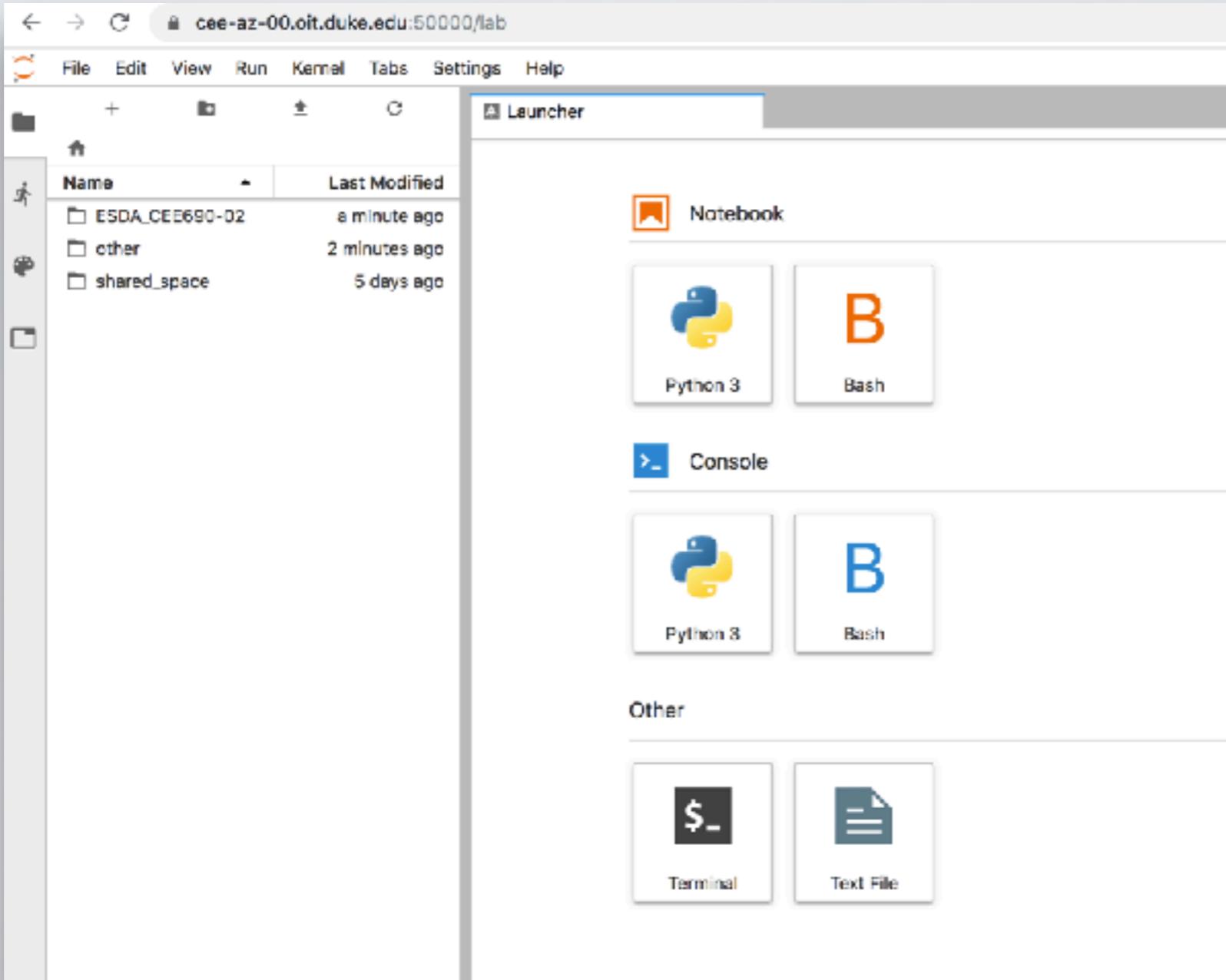
Course Information

Lectures are on Tuesdays and Thursdays from 3:05 PM - 4:20 PM in Teer 106. The course website is on GitHub (<https://github.com/chaneyn/CEE690-02>). Class announcements will be made via Sakai (CEE690.02.F19).

Instructor

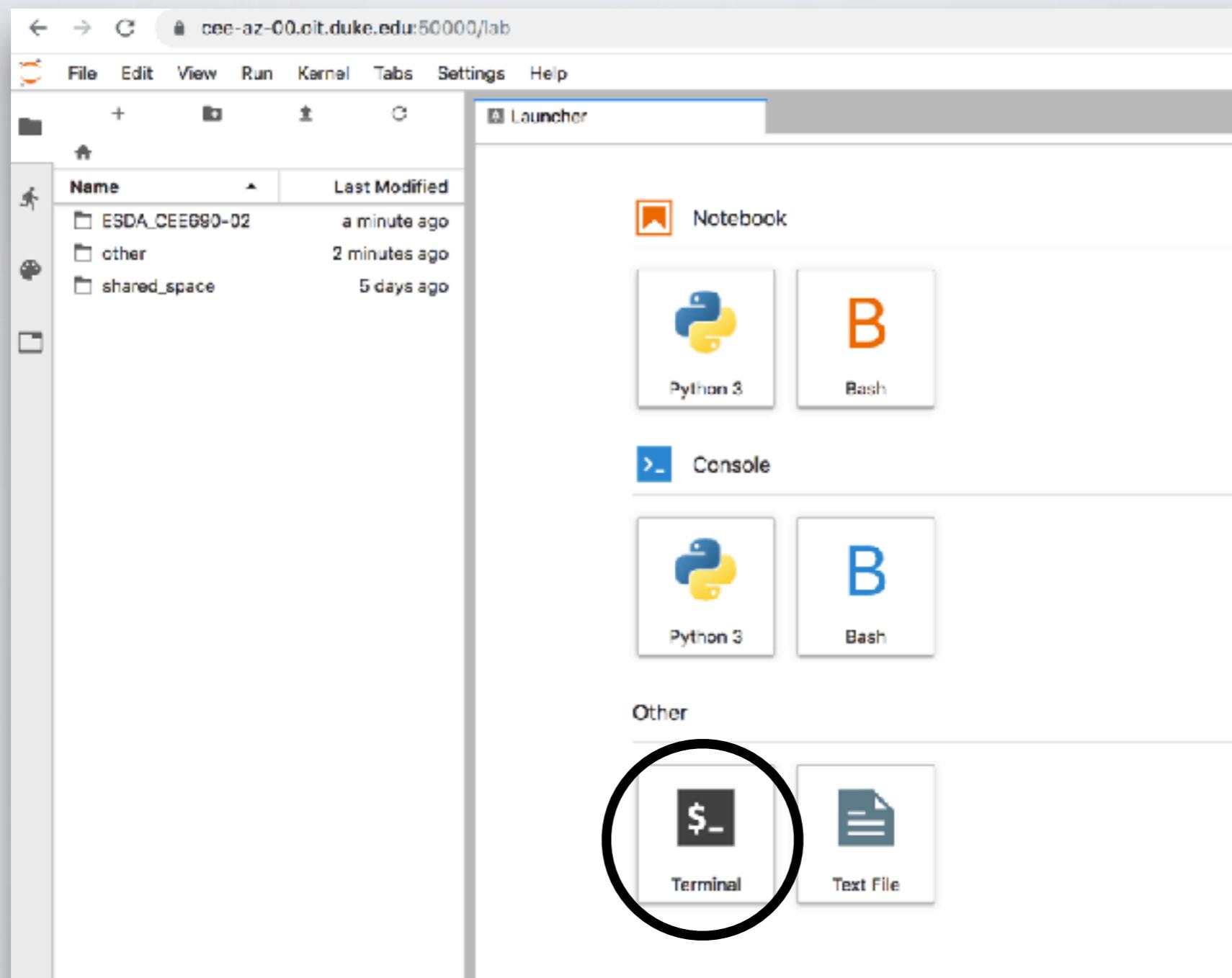
Professor Nathaniel W. Chaney (Nate)
Email: nathaniel.chaney@duke.edu
Office: FCIEMAS 2463
Office hours: By appointment on Friday mornings (calendly.com/nathaniel-chaney/)

Getting started: Clone the course's GitHub repository



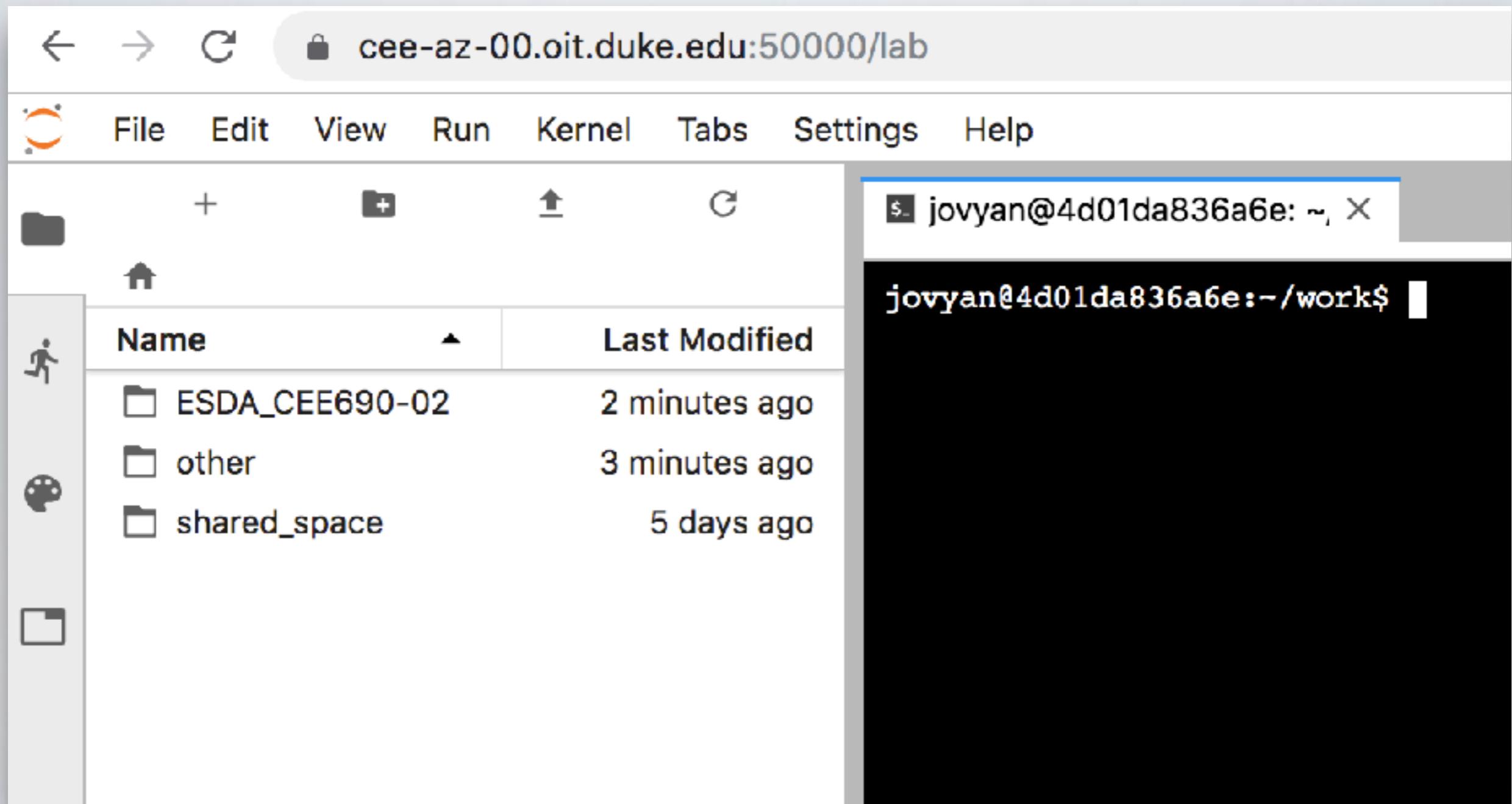
Navigate to your Docker container instance

Open a terminal !

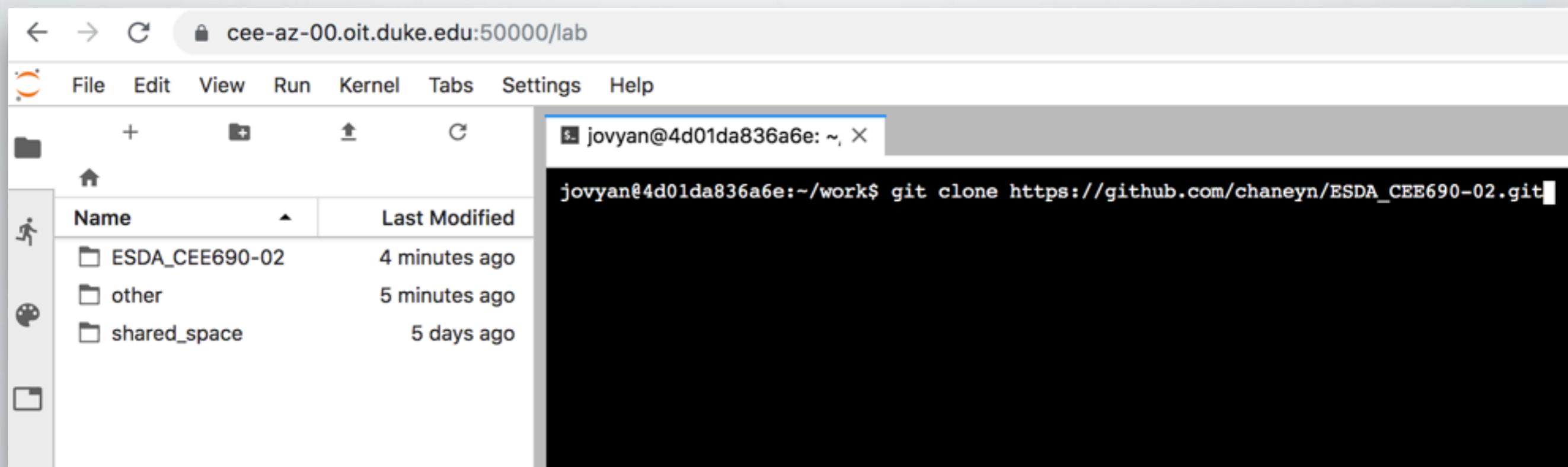


Press the terminal button

Open a terminal II



Write the following git command

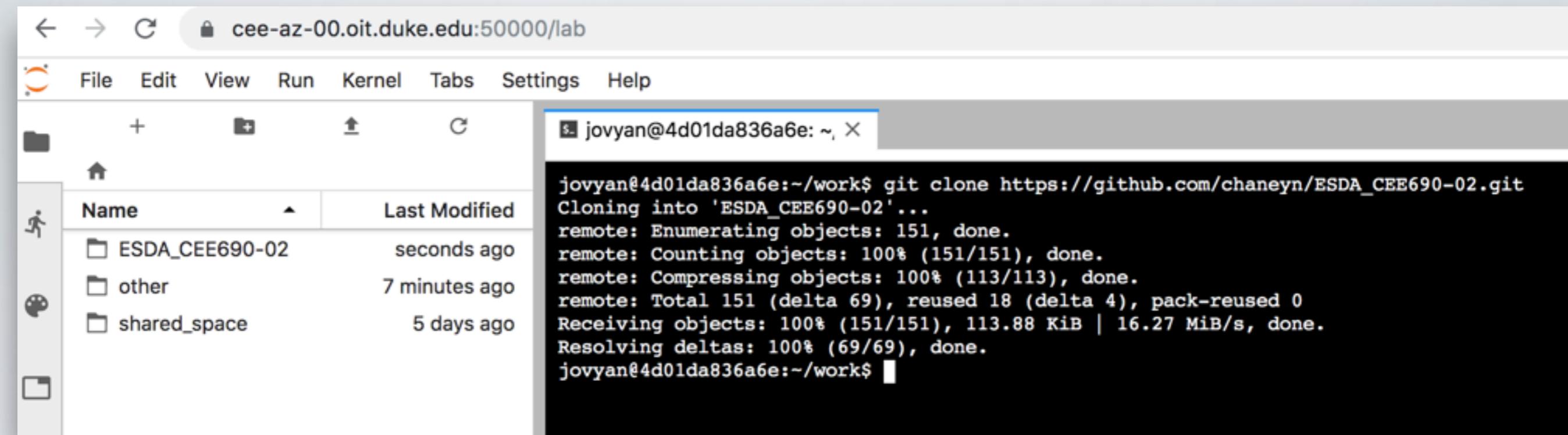


A screenshot of a terminal window titled "jovyan@4d01da836a6e: ~, X". The window shows a command being typed: "jovyan@4d01da836a6e:~/work\$ git clone https://github.com/chaneyn/ESDA_CEE690-02.git". The terminal is part of a desktop environment with a file manager sidebar on the left.

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.

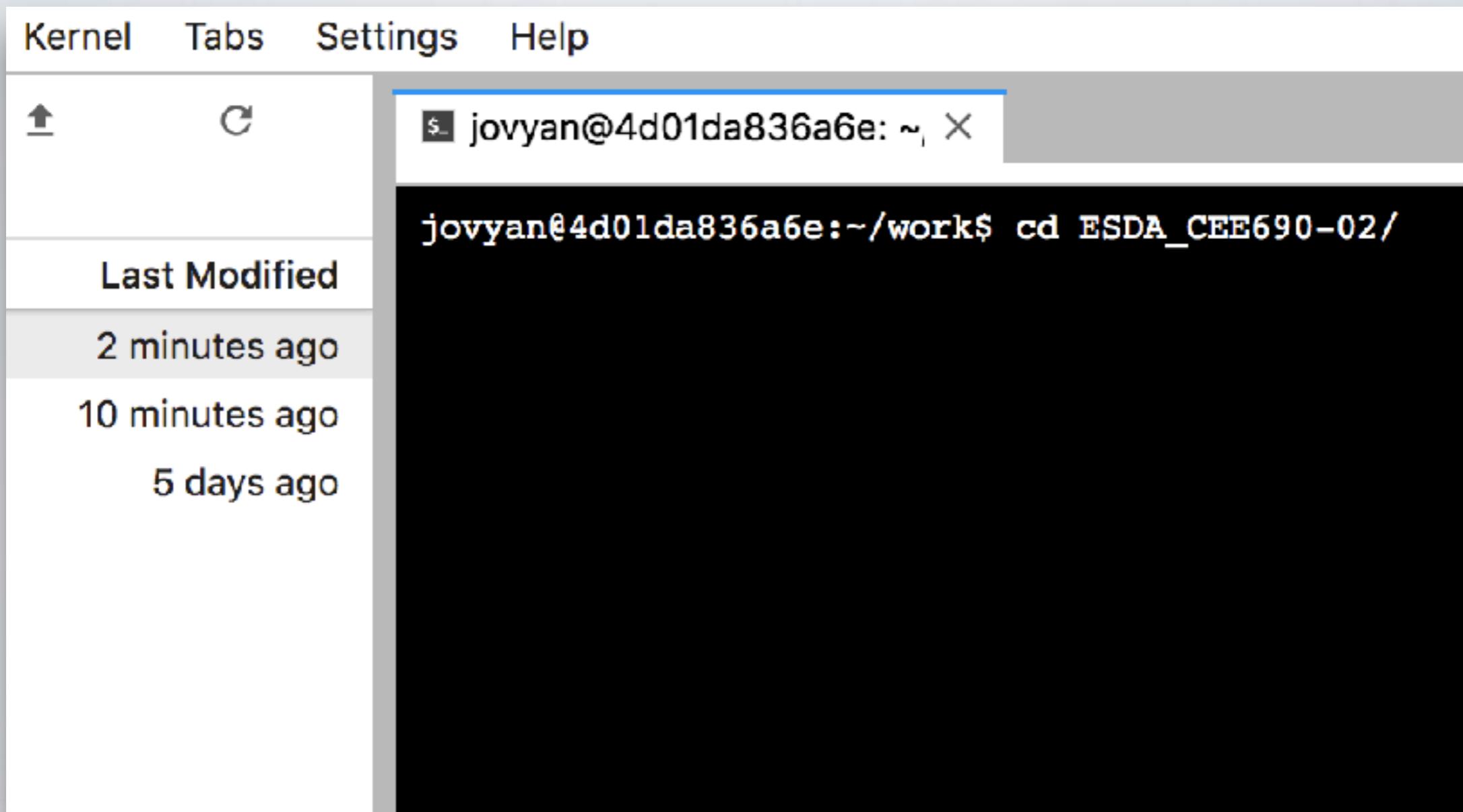
## Instructor
Professor Nathaniel W. Chaney (Nate)
Email: nathaniel.chaney@duke.edu
Office: FCIEMAS 2463
Office hours: By appointment on Friday mornings (calendly.co/nathanielchaney)

## 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 with the prompt "\$ jovyen@4d01da836a6e: ~, X". Below the prompt, the command "jovyen@4d01da836a6e:~/work\$ cd ESDA_CEE690-02/" is typed. The terminal background is black, and the text is white.

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":

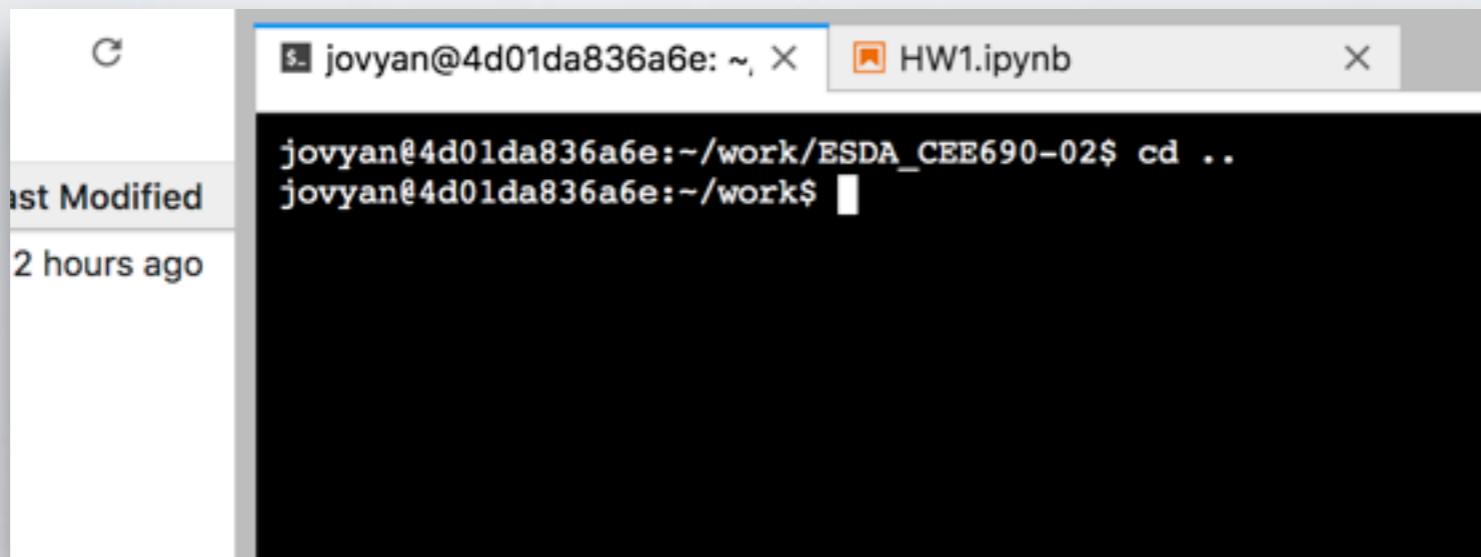
- 3 minutes ago
- 1 minutes ago
- 5 days ago

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 on a Linux system, indicated by the prompt "jovyan@4d01da836a6e:~/" and the command "cd ..". The window has a dark background and light-colored text.

**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 .
```

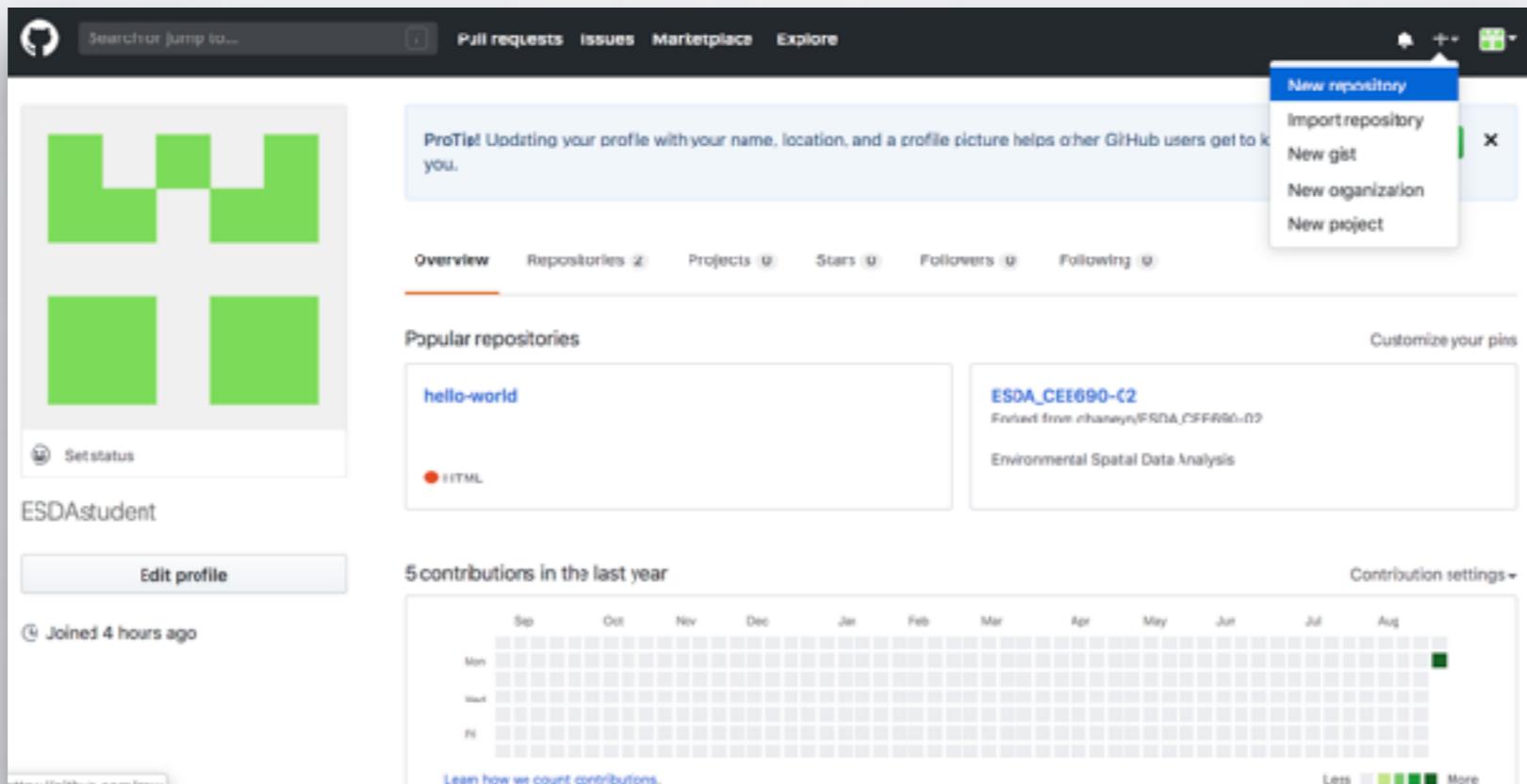
Add assignment to your repository

```
:~/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



Match names

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_nwc 

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 ▾

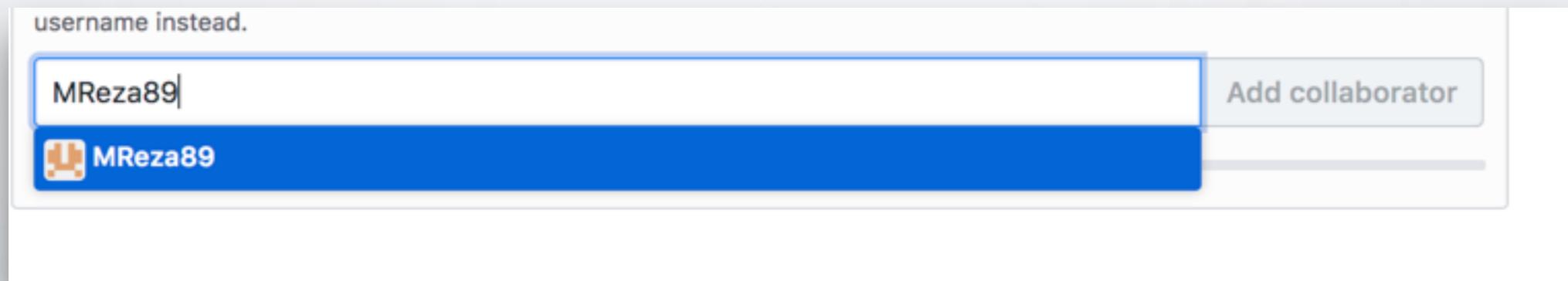
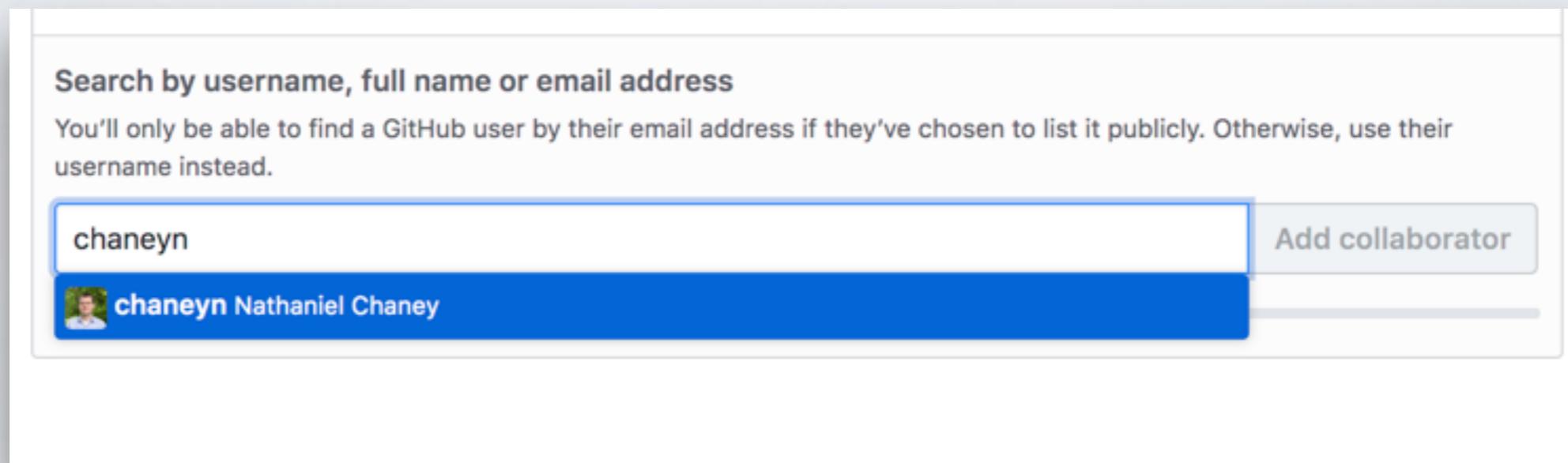
chaneyn Added HW1 file Latest commit 7f144f0 8 minutes ago

HW1.ipynb Added HW1 file 8 minutes ago

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

Last thing to do is to share your private repository with Reza and Nate

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.