Introduction to Data Science Software and Platforms

8/24/20

Data Science Software and Platforms we will Use

Main Purpose in Class	Software/Platform	What is it?
Coding Lab Assignments in Python	**Anacondas**	**Distribution** of the Python and R programming languages. Allows you to download and run popular Python packages and the Jupyter Notebook Application.
	Jupyter Notebooks	**Python application** that allows you to write data science reports that also need to be integrated with interactive Python code blocks.
	Python	**A programming language**
Version Control: practice of tracking and managing changes to code.	**Git**	Version control **system**.
	Github	Git repository **hosting service**.
	Github Enterprise STAT207 Organization https://github-dev.cs.illinois.edu/stat207-fa20	**A collection of user accounts** (users=you, your classmates, Dr. Ellison, TAs, and Cas) **that owns Github repositories.**
	Command Line Interface	An **application** that processes commands to a computer program in lines of text.

Downloading a Version of Anaconda (Windows)

Installing Python

You will need Python 3.6 (or later). We will first check if you have Python already (if you have done Data Science) and install it if you don't already have it.

Checking for existing Python

1. Open up your command prompt
2. Type python --version and press **Enter**.

- If you see Python 3.7.1 (or similar), you are all set – no need to install Python. *(Skip to the git section.)*
- If you see 'python' is not recognized as an internal or external command, operable program or batch file., install it now:

Installing Python

1. Visit https://conda.io/miniconda.html to get Miniconda, a light-weight version of the python programming language
2. Download and install the latest Windows, **64-bit** installer for the latest version of Python (eg: 3.7).
3. After the install finishes, exit your command prompt, re-launch it, and verify it installed by following the steps above (in "checking for existing python").

http://courses.las.illinois.edu/fall2020/stat207/datascience-setup.html

Downloading a Version of Anaconda (MAC OS X)

Installing Python

You will need Python 3.6 (or later). We will first check if you have Python already (if you have done Data Science) and install it if you don't already have it.

Checking for existing Python

1. Open up your command prompt
2. Type python --version and press **Enter**.

- If you see Python 3.7.1 (or similar), you are all set – no need to install Python!
- If you see an error or Python 2.7, we will install it now!

Installing Python

1. Visit https://conda.io/miniconda.html to get Miniconda, a light-weight version of the python programming language
2. Download the latest Mac OS X, **64-bit bash** installer for the latest version of Python (eg: 3.7).
3. Open up your command prompt and run the script you downloaded by running the following:

```
cd Downloads

bash Miniconda3-latest-MacOSX-x86_64.sh
```

You will need to press q to exit the license screen and all default options are fine.

1. Restart your terminal

http://courses.las.illinois.edu/fall2020/stat207/datascience-setup.html

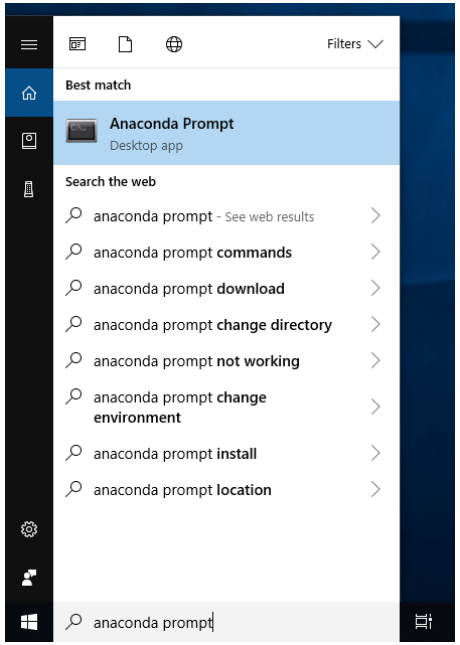
Anaconda

(What applications it comes with)

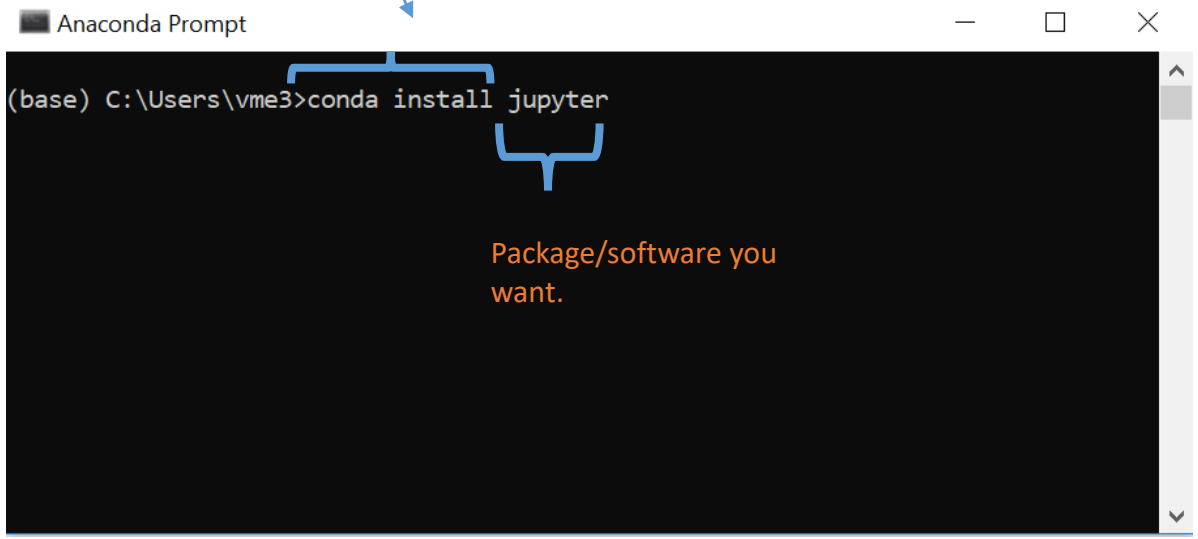
All versions of Anaconda should come with an **Anaconda Prompt**, a command line interface that allows you to run python commands.

Anaconda Prompt commonly used for:

- Installing python packages and applications
- *One way* to launch python applications.

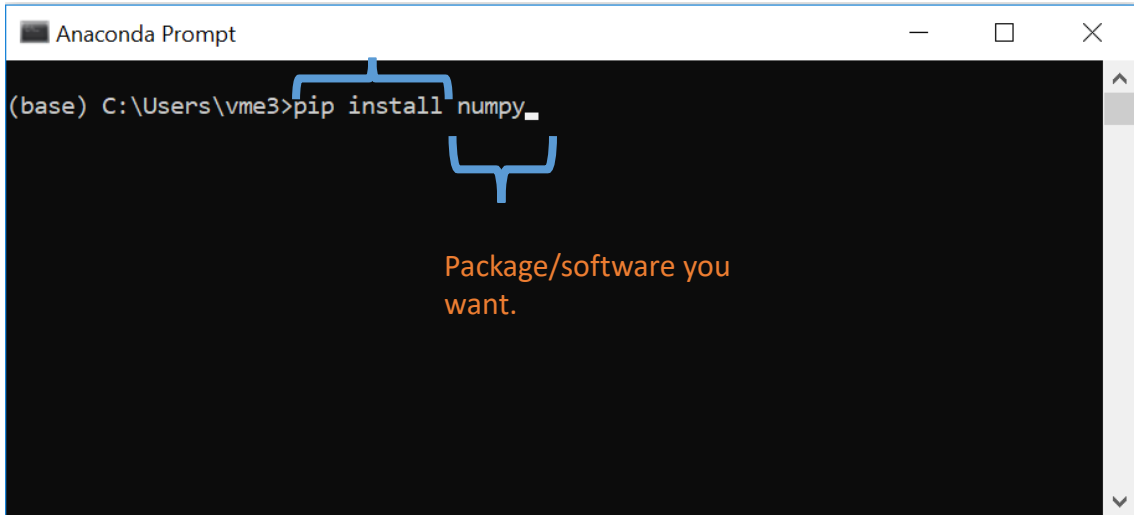


Code for installing *some* software/packages.



Package/software you want.

Code for installing *some* software/packages.



Package/software you want.

Downloading Jupyter Notebooks and Other Python Packages/Applications

Part 1c: Set up your Python notebook

In Data Science, all of our programming will be done in “notebooks”. Your python install will need a few **libraries** in order to run the notebooks. Using your command line, run the following:

```
conda install jupyter
conda install pandas
conda install matplotlib
conda install seaborn
```

Potential Error Workaround: IF you get an error about "conda not found" when trying to do this, you can also install these packages by doing the following.

- Searching for the "miniconda" program you just downloaded, and run what should say "Anaconda Prompt."
- This will open up another command line window that is specifically for running python commands (for instance commands that install packages).
- Run the code in this Anaconda Prompt instead

  ```
  conda install jupyter
  conda install pandas
  conda install matplotlib
  conda install seaborn
  ```

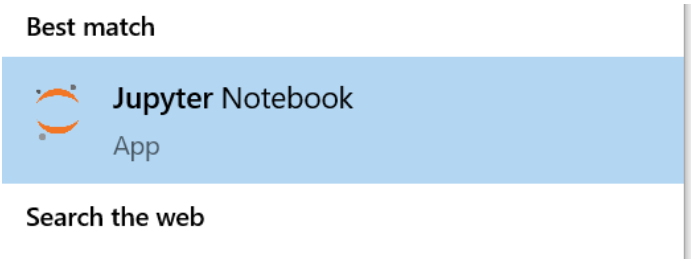
This might take a couple of minutes. You will need to type [y] to confirm you want to install of of the packages (the option [y]/n shows that y is default when you choose no option).

You can check what has been installed already using the command:

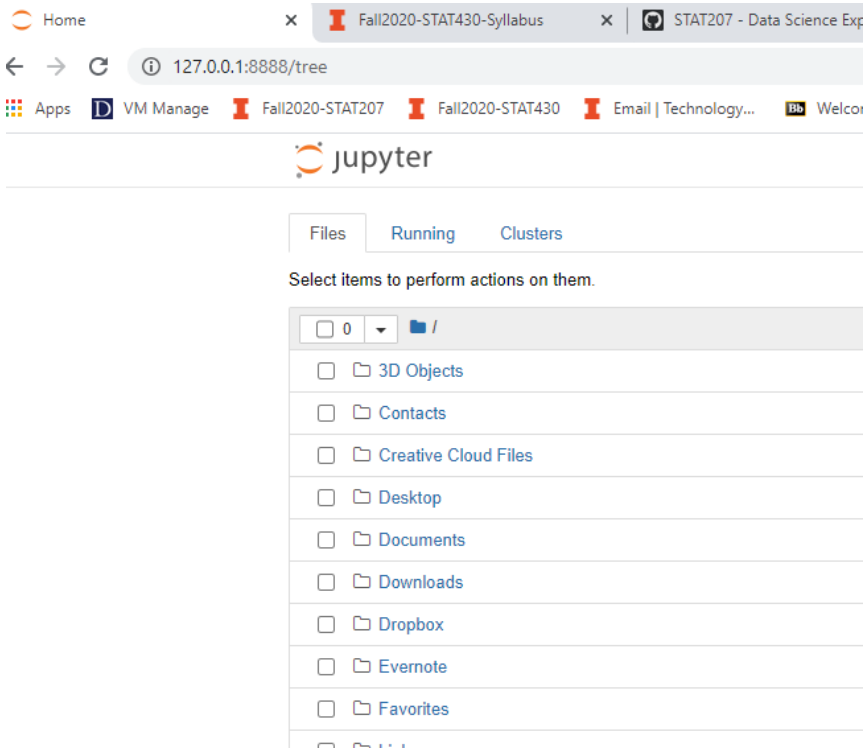
```
conda list
```

http://courses.las.illinois.edu/fall2020/stat207/labs/01-intro.html

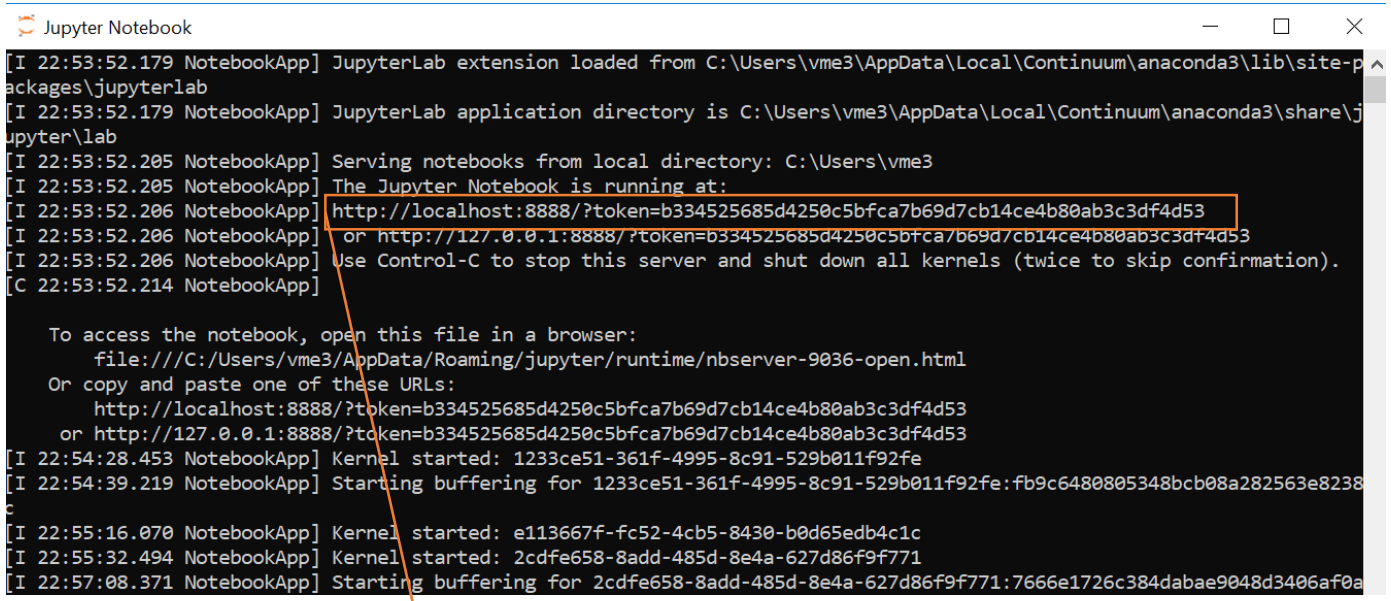
Running Jupyter Notebooks



Displays a File System of your Computer in a Browser



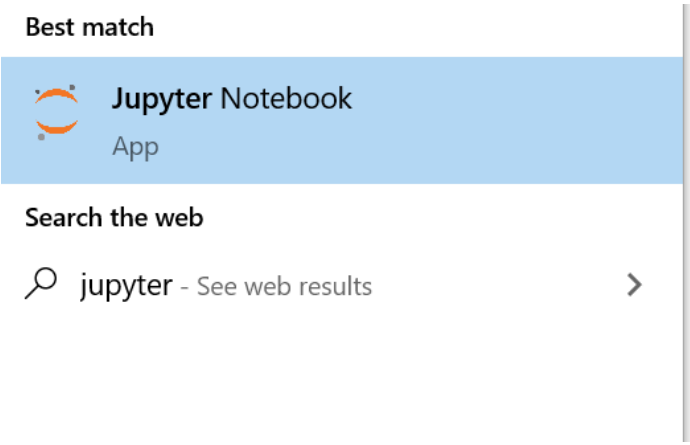
Launches the Notebook Application with Information about your Jupyter Notebook Session



Url to relaunch the Jupyter Notebook your browser.

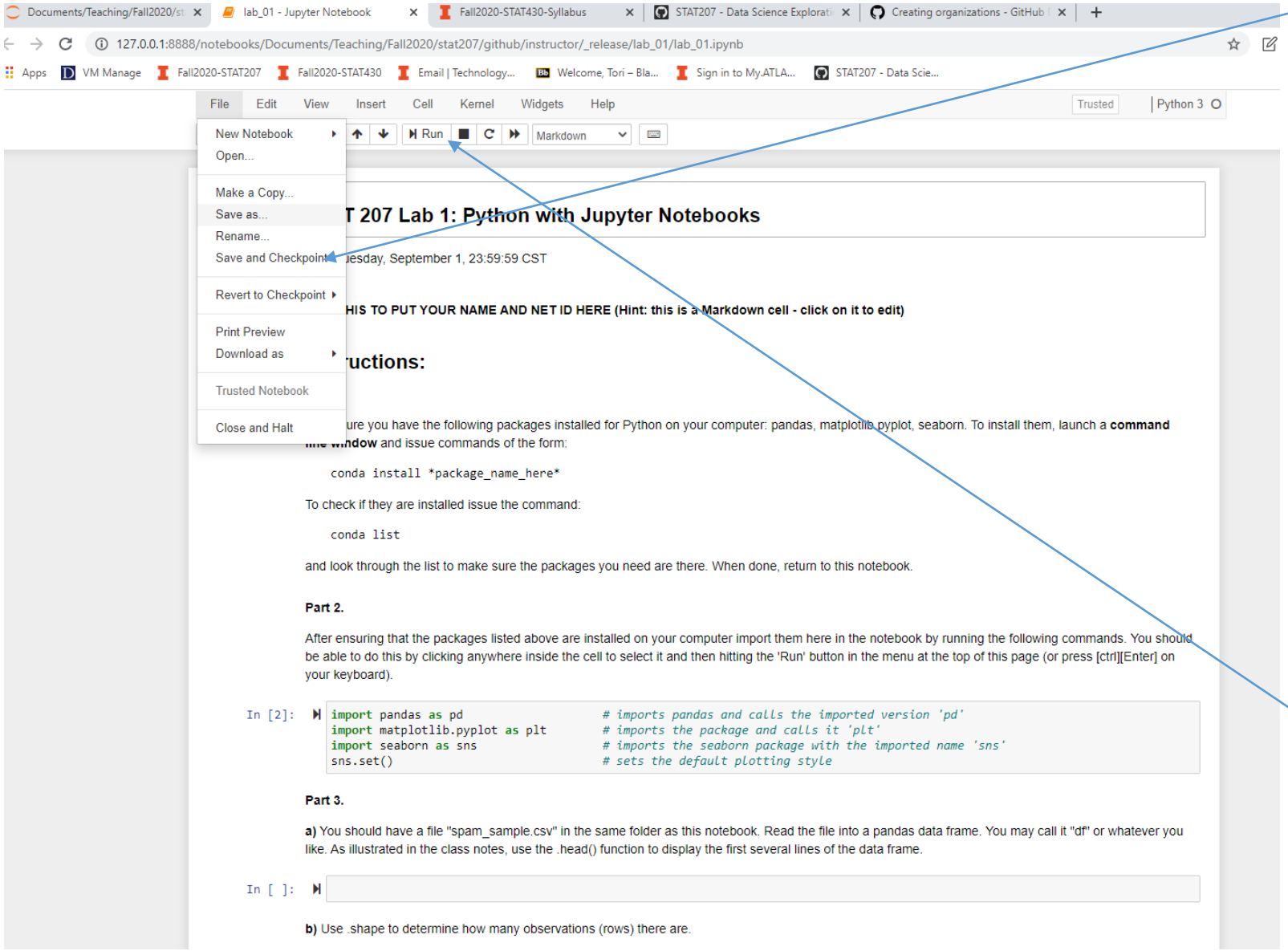
…navigate to where your notebooks are saved!

Running Jupyter Notebooks



Open your notebook

Running Jupyter Notebooks



Don't forget to **save** your work before exiting!

[ctrl][s]

To **run code**, click the code block and then either do:

- [ctrl][Enter] or
- Click run.

Installing Git (Windows)

Installing Git

Any modern version of git works. We will first check if you have git and install it if you don't already have it.

Checking for git

1. Open up your command prompt
2. Type git --version and press **Enter**.

- If you see git version ... (or similar), you are all set – no need to install git! *(You're done!)*
- If you see 'git' is not recognized as an internal or external command, operable program or batch file., install it now:

Installing git

1. Visit https://git-scm.com/downloads to get git, a distributed version control system/repository tool
2. Download and install the latest Windows installer. (You should not need to select/unselect any of the options that are already preselected in the installation proces... aka just keep hitting next.)
3. After the install finishes, exit your command prompt, re-launch it, and verify it installed by following the steps above.

http://courses.las.illinois.edu/fall2020/stat207/datascience-setup.html

Installing Git (Mac OS X)

Installing Git

Any modern version of git works. We will first check if you have git and install it if you don't already have it.

Checking for git

1. Open up your command prompt
2. Type git --version and press **Enter**.

- If you see git version ... (or similar), you are all set – no need to install git!
- If you see an error, we will install it now!

Installing git

1. Visit https://git-scm.com/downloads to get git, a distributed version control system/repository tool
2. Download and install the latest Mac OS X installer.
3. After the install finishes, verify it installed by following the steps above.

http://courses.las.illinois.edu/fall2020/stat207/datascience-setup.html

Setting up a Git Repository in our STAT207 Github Enterprise Organization https://github-dev.cs.illinois.edu/stat207-fa20

Part 0: General Class Folder

First, you should create a folder named 'stat207' (we recommend on your Desktop) to hold all of your Python notebooks.

Guide: Setting Up git for Data Science Exploration

To set up git, there are certain commands you will run:

1. Once for the entire semester ("Course Setup"),
2. Once for each computer you use ("Computer Setup"),
3. Once each time you work on Data Science Discovery ("Assignment Setup")

Course Setup

To begin to work on assignments and turn in work, you will need to create a git repository for the course.

- Visit: https://edu.cs.illinois.edu/create-ghe-repo/stat207-fa20/
- Follow the instructions to create your repository, coming back here once you have the URL

Computer Setup

1. Create a stat207 folder on your Desktop (if you haven't already)
2. Using your command line, run the following commands to navigate into your stat207 folder:

- cd Desktop
- cd stat207

3. **Clone** a local copy of your git repository with the following command (making sure to replace YOUR-NETID with your own):

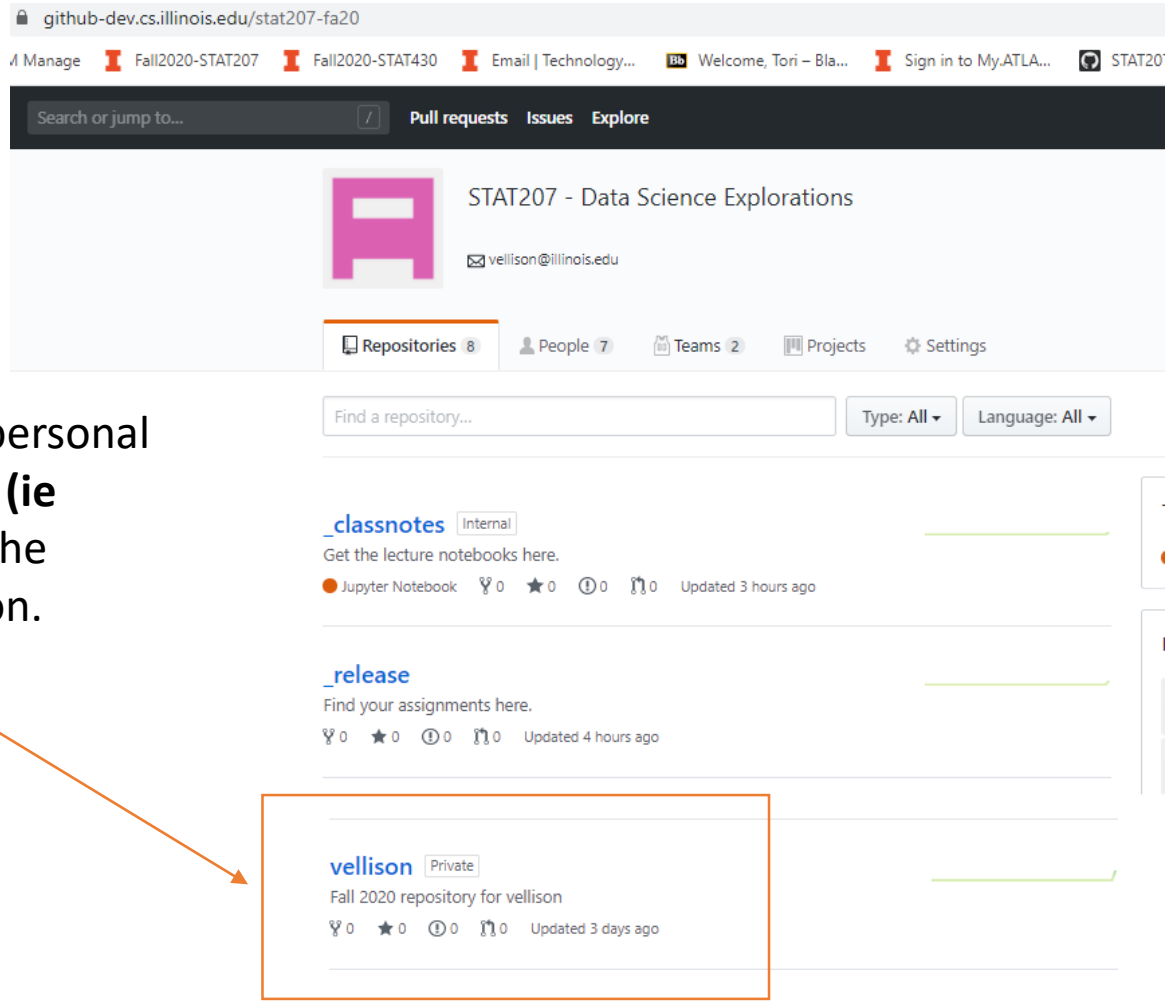
- git clone https://github-dev.cs.illinois.edu/stat207-fa20/YOUR_NETID
- You may have to enter your netid/password

4. Navigate into your git directory by going into your NetID-named folder (replace NETID with yours):

- cd NETID

5. Set up a connection to the _release repository where code will be released for you:

- git remote add release https://github-dev.cs.illinois.edu/stat207-fa20/_release.git



Creates a personal **repository (ie folder)** in the organization.

Setting up a Git Repository in our STAT207 Github Enterprise Organization https://github-dev.cs.illinois.edu/stat207-fa20

Part 0: General Class Folder

First, you should create a folder named 'stat207' (we recommend on your Desktop) to hold all of your Python notebooks.

Guide: Setting Up git for Data Science Exploration

To set up git, there are certain commands you will run:

1. Once for the entire semester ("Course Setup"),
2. Once for each computer you use ("Computer Setup"),
3. Once each time you work on Data Science Discovery ("Assignment Setup")

Course Setup

To begin to work on assignments and turn in work, you will need to create a git repository for the course.

- Visit: https://edu.cs.illinois.edu/create-ghe-repo/stat207-fa20/
- Follow the instructions to create your repository, coming back here once you have the URL

Computer Setup

1. Create a stat207 folder on your Desktop (if you haven't already)
2. Using your command line, run the following commands to navigate into your stat207 folder:

- cd Desktop
- cd stat207

3. **Clone** a local copy of your git repository with the following command (making sure to replace YOUR-NETID with your own):

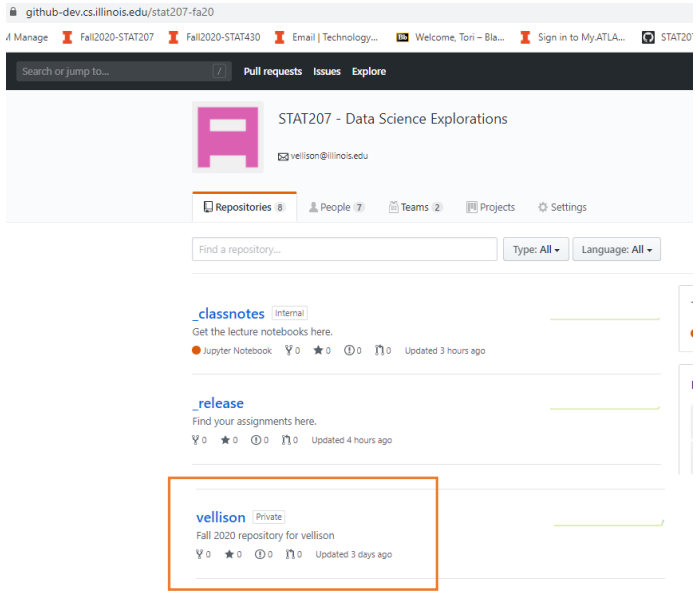
- git clone https://github-dev.cs.illinois.edu/stat207-fa20/YOUR_NETID
- You may have to enter your netid/password

4. Navigate into your git directory by going into your NetID-named folder (replace NETID with yours):

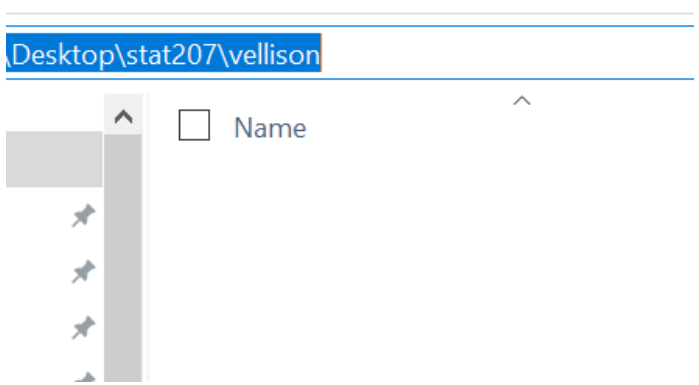
- cd NETID

5. Set up a connection to the _release repository where code will be released for you:

- git remote add release https://github-dev.cs.illinois.edu/stat207-fa20/_release.git

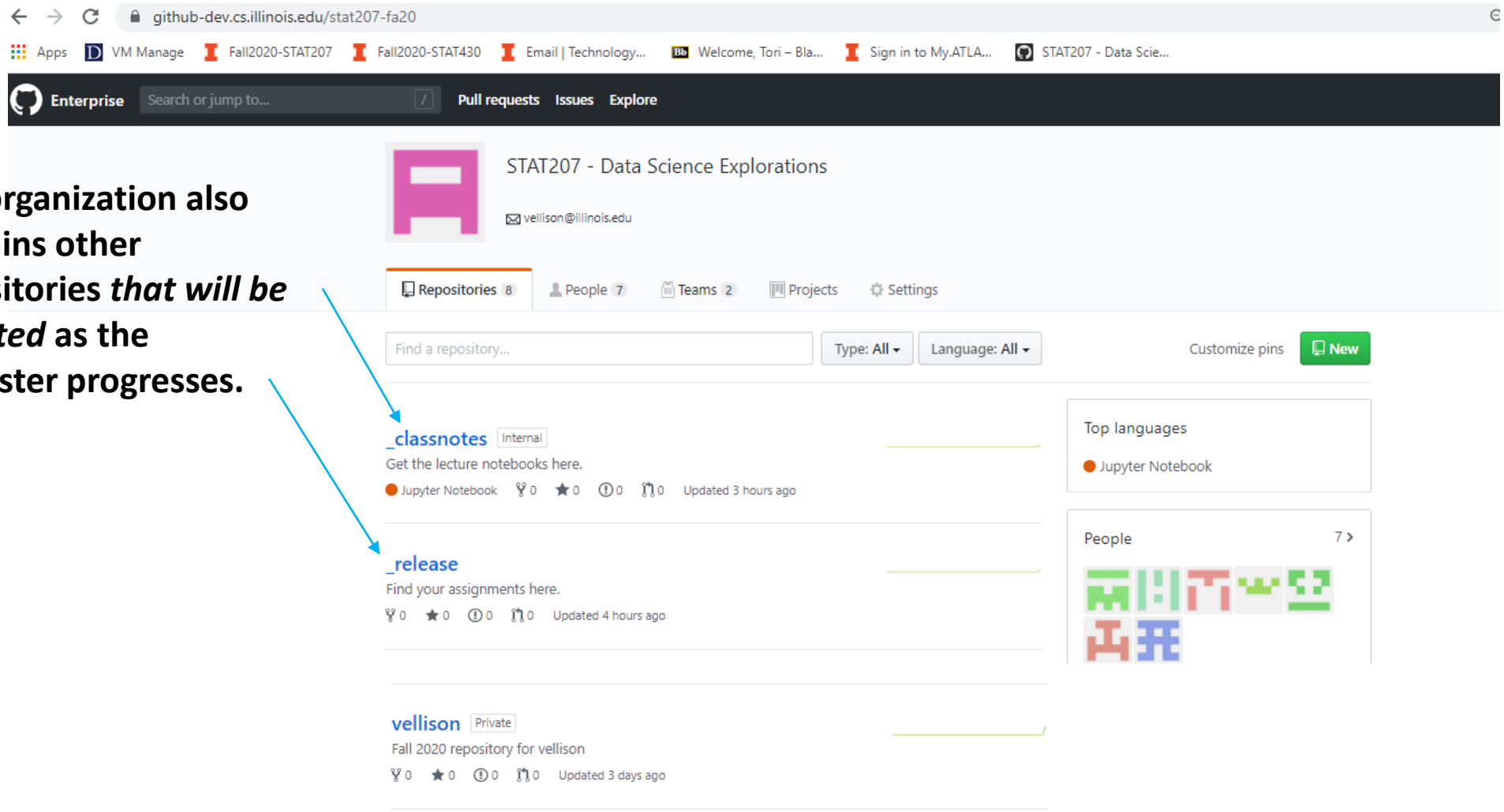


Only done once: Clones your personal repository to your local computer.

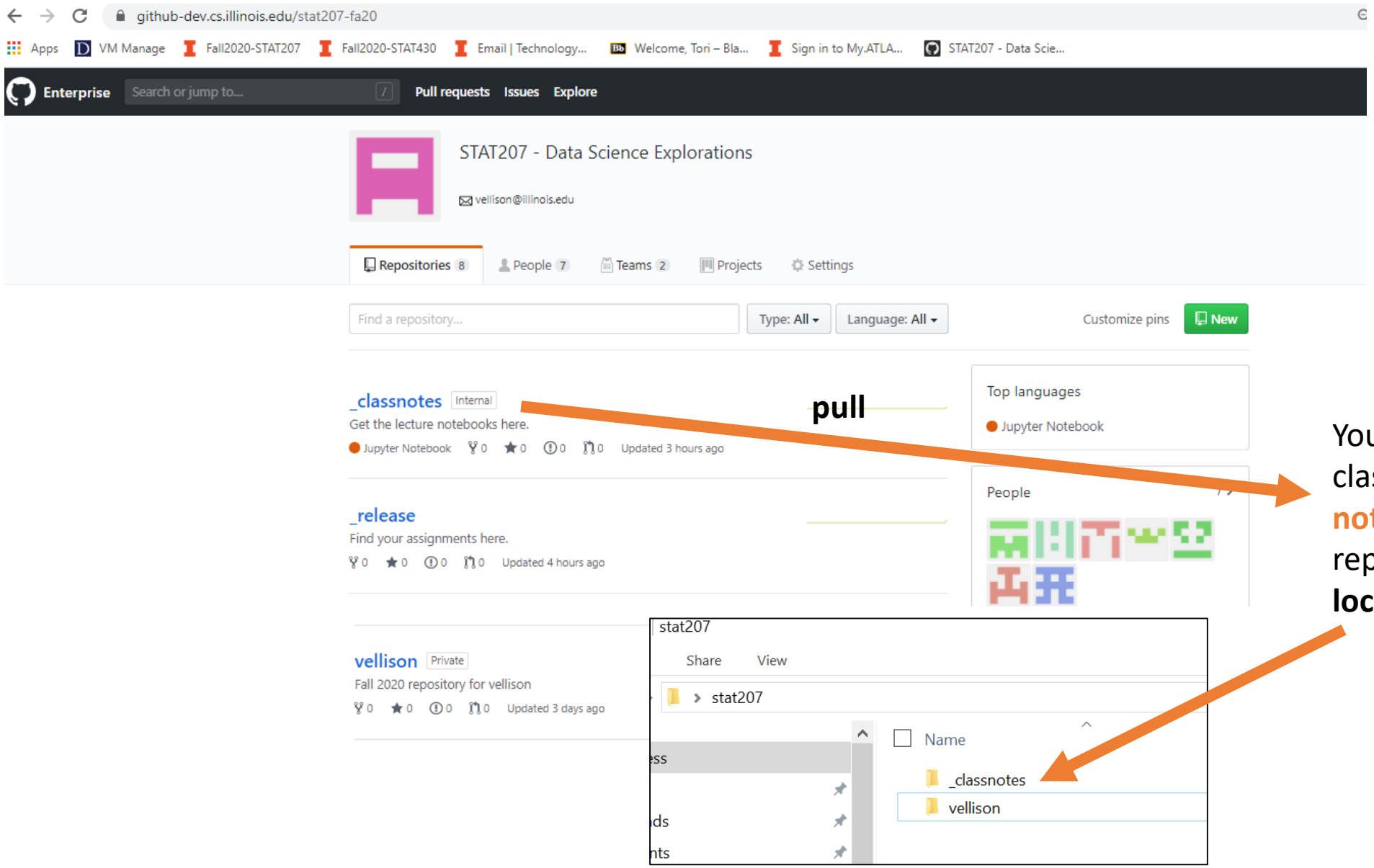


Our STAT207 Github Enterprise Organization

Our organization also contains other repositories *that will be updated* as the semester progresses.



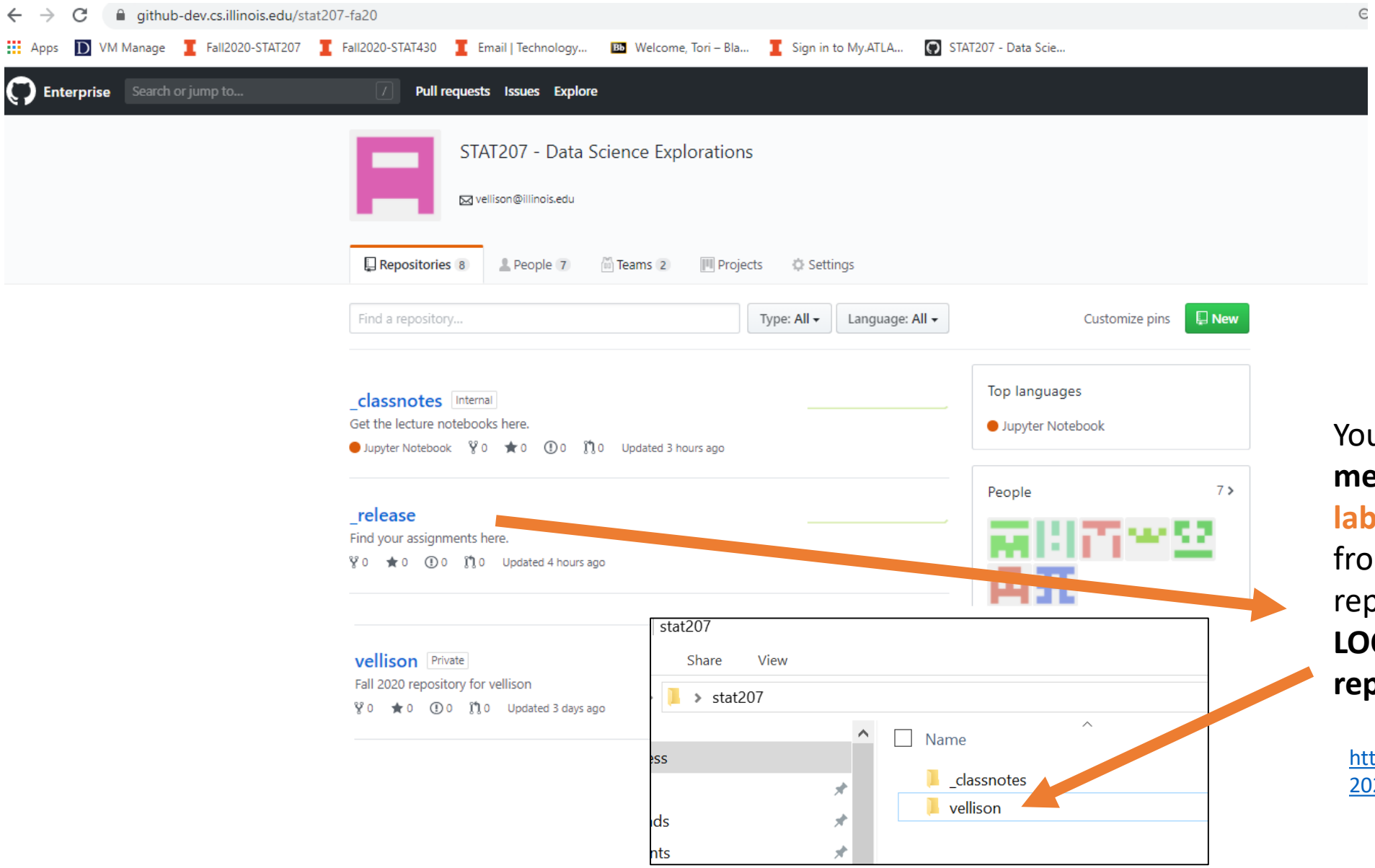
Our STAT207 Github Enterprise Organization



http://courses.las.illinois.edu/fall2020/stat207/classnotes.html

You will **pull** the class **lecture notes** from this repository to your **local computer**.

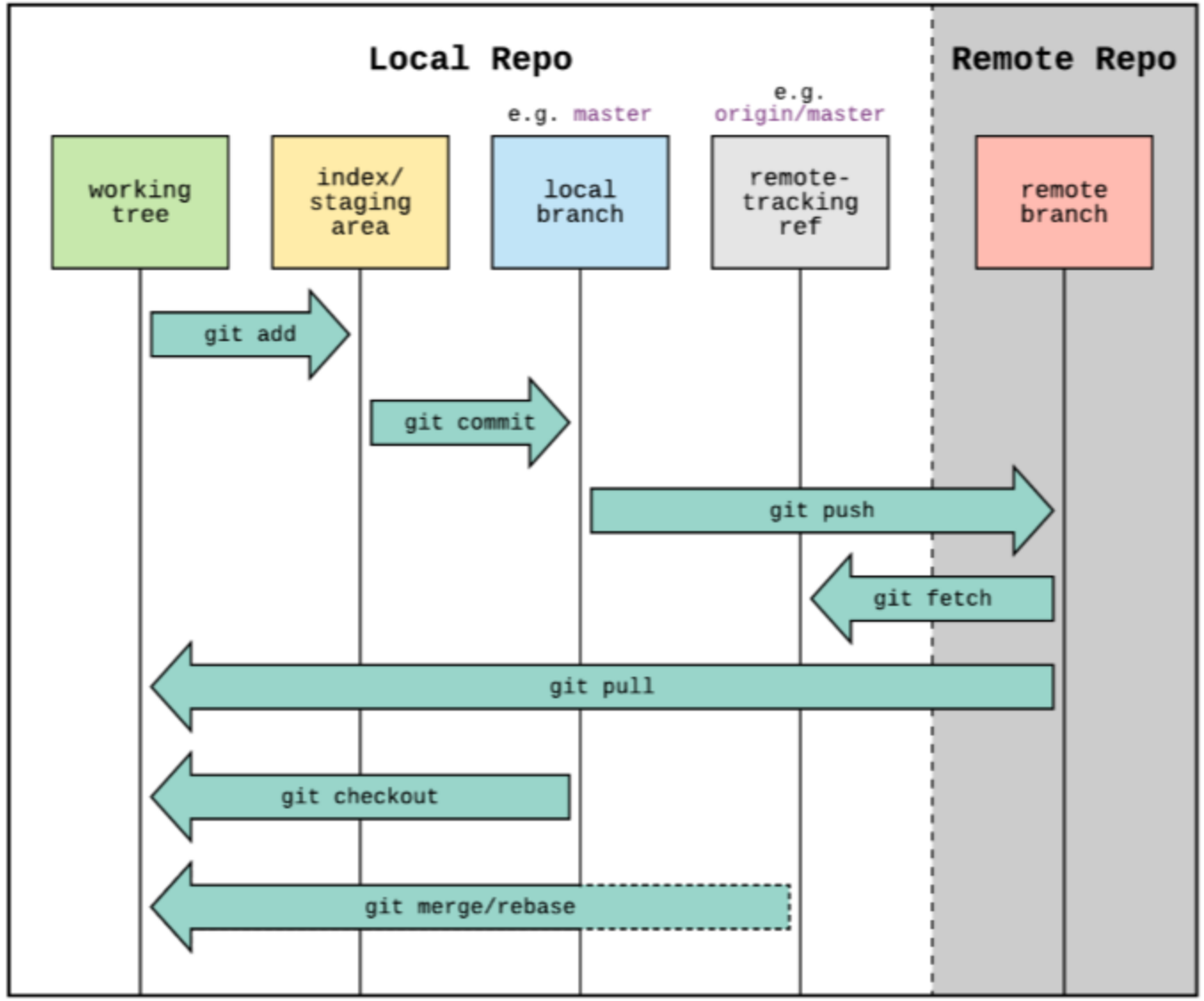
Our STAT207 Github Enterprise Organization



You will **fetch** and **merge** the class **lab assignments** from this repository to your **LOCAL netid repository.**

http://courses.las.illinois.edu/fall2020/stat207/labs/01-intro.html

Git Workflow, what do these **git commands mean?**

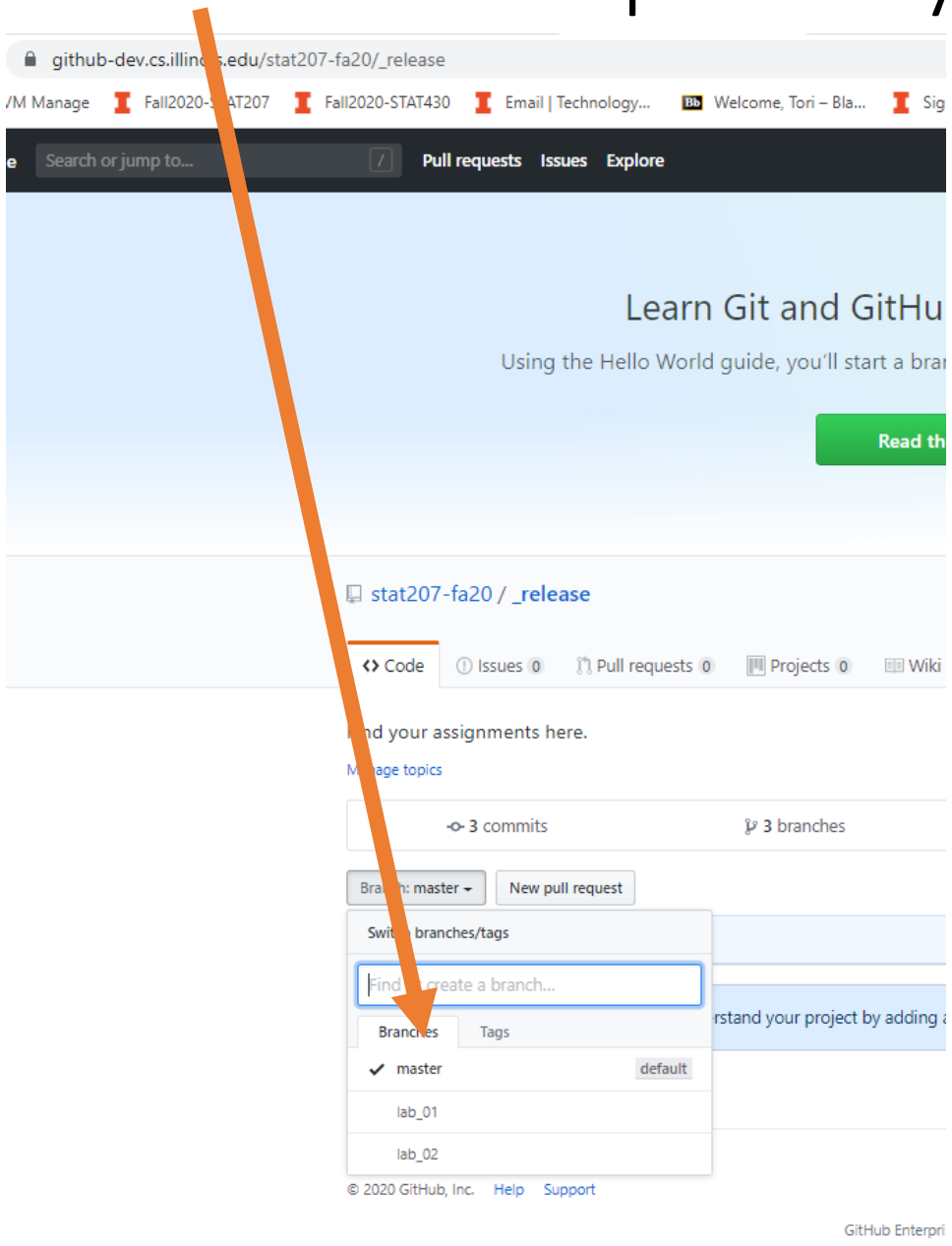


https://www.javatpoint.com/git-version-control-system

Commands:

- **git fetch release**
  - Transmits the "whole version" of a REMOTE repository to your LOCAL computer.
- **git merge release/lab_01 –m "merging initial files"**
  - Merges just the changes made to the lab_01 **branch** of the remote release repository.

Branches of a Repository



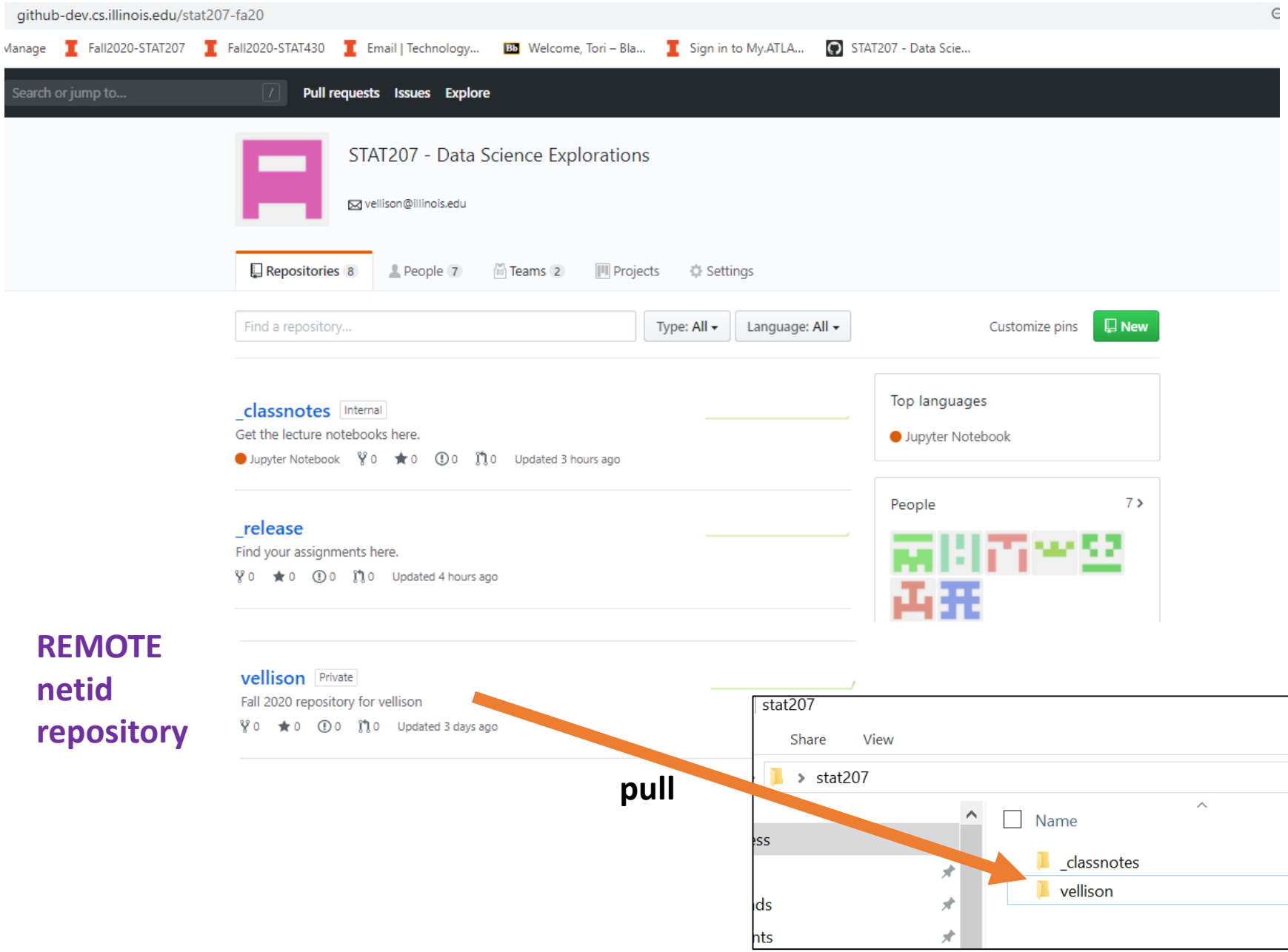
- Repositories can have multiple **branches** (or versions).

Our STAT207 Github Enterprise Organization

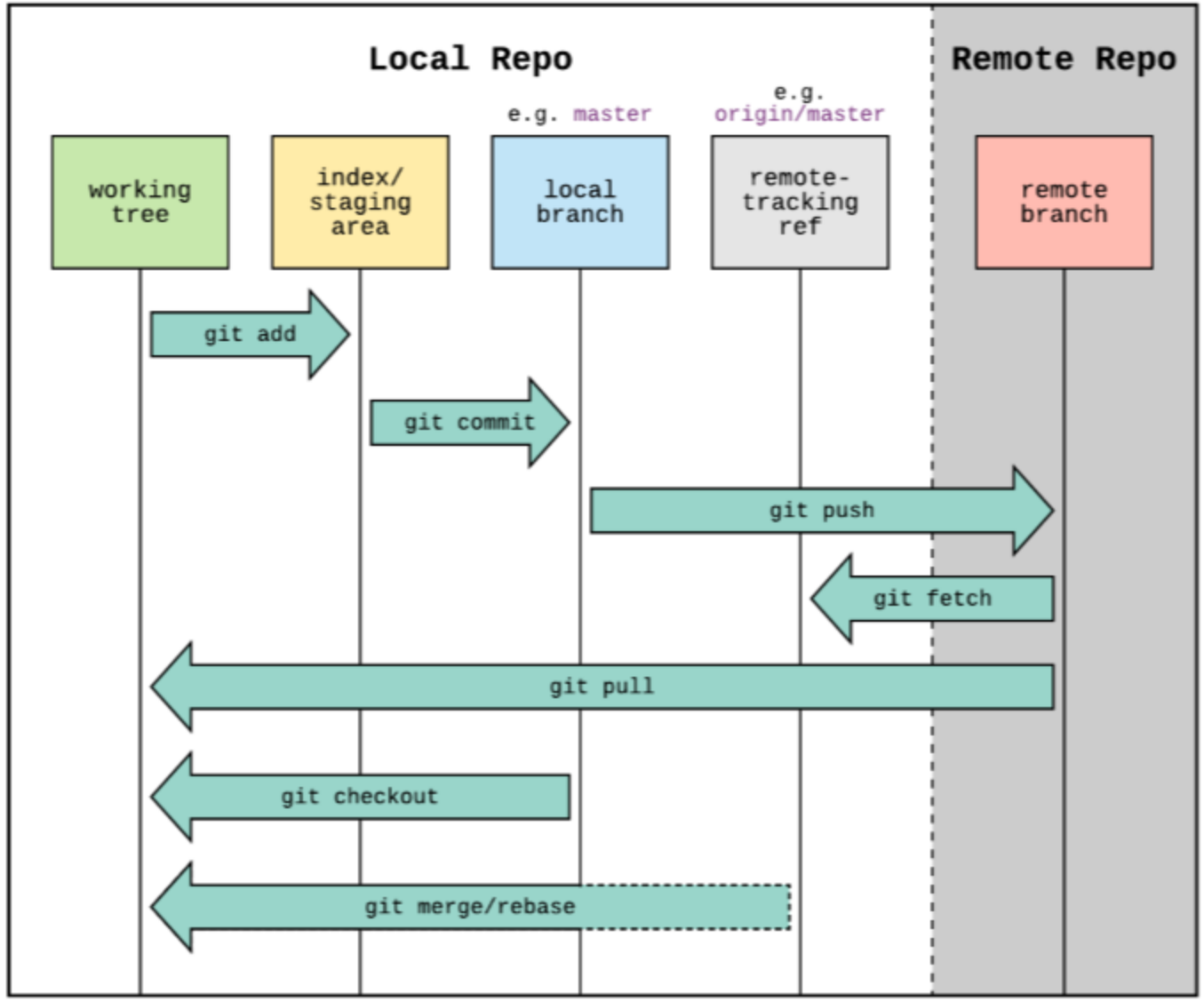
http://courses.las.illinois.edu/fall2020/stat207/labs/01-intro.html

After editing the contents of your **local *current version* of your netid repository (like your lab assignments)** you will do the following to submit them for saving/grading.

- **pull** any changes from your **REMOTE netid repository** to your **LOCAL netid repository**



Git Workflow, what do these **git commands mean?**



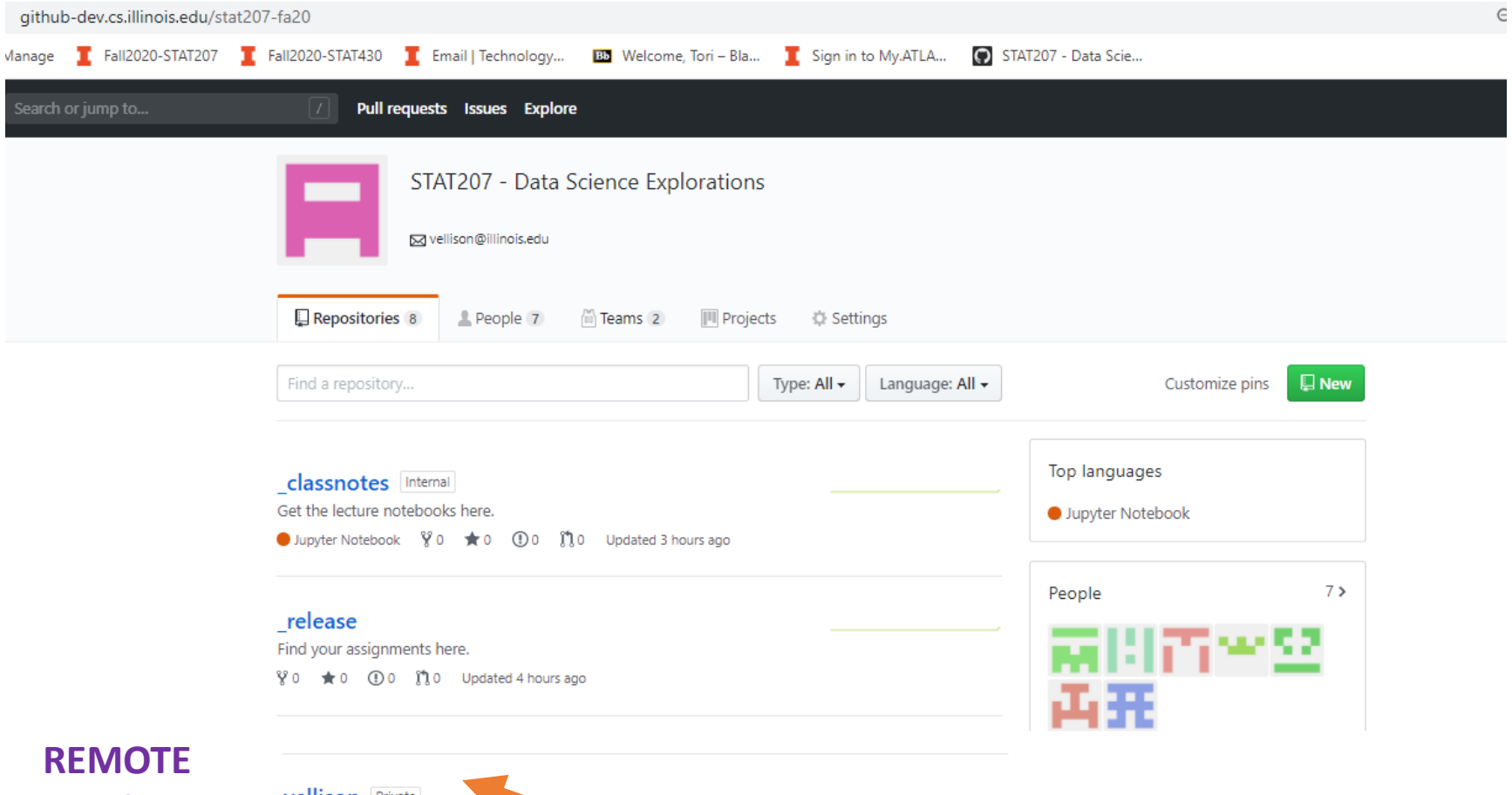
https://www.javatpoint.com/git-version-control-system

Commands:

- **git pull**
  - Pulls in (and automatically merges) any changes from the REMOTE repository into the LOCAL repository.

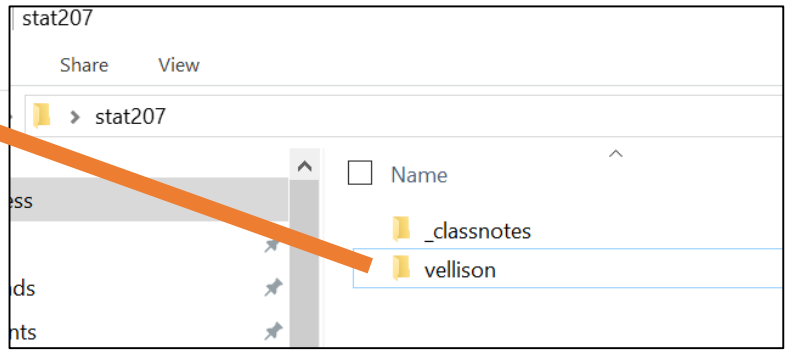
Our STAT207 Github Enterprise Organization

http://courses.las.illinois.edu/fall2020/stat207/labs/01-intro.html



REMOTE netid repository

add, commit, and **push**

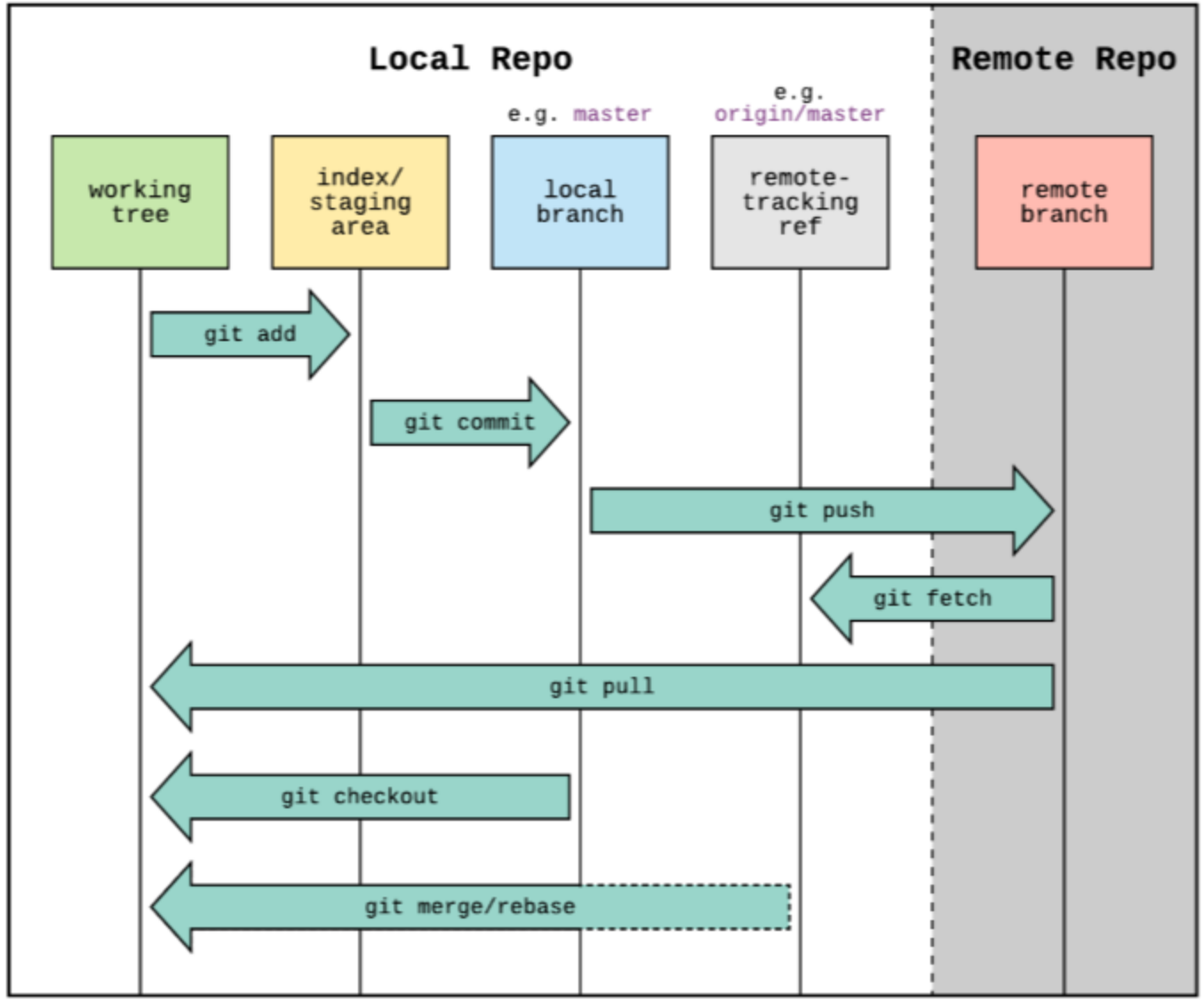


LOCAL netid repository

After editing the contents of your **local *current version* of your netid repository (like your lab assignments)** you will do the following to submit them for saving/grading.

- **pull** any changes from your **REMOTE netid repository** to your **LOCAL netid repository**
- And then **add, commit,** and **push** the changes from your LOCAL netid repository to your REMOTE netid repository for grading/saving.

Git Workflow, what do these **git commands mean?**



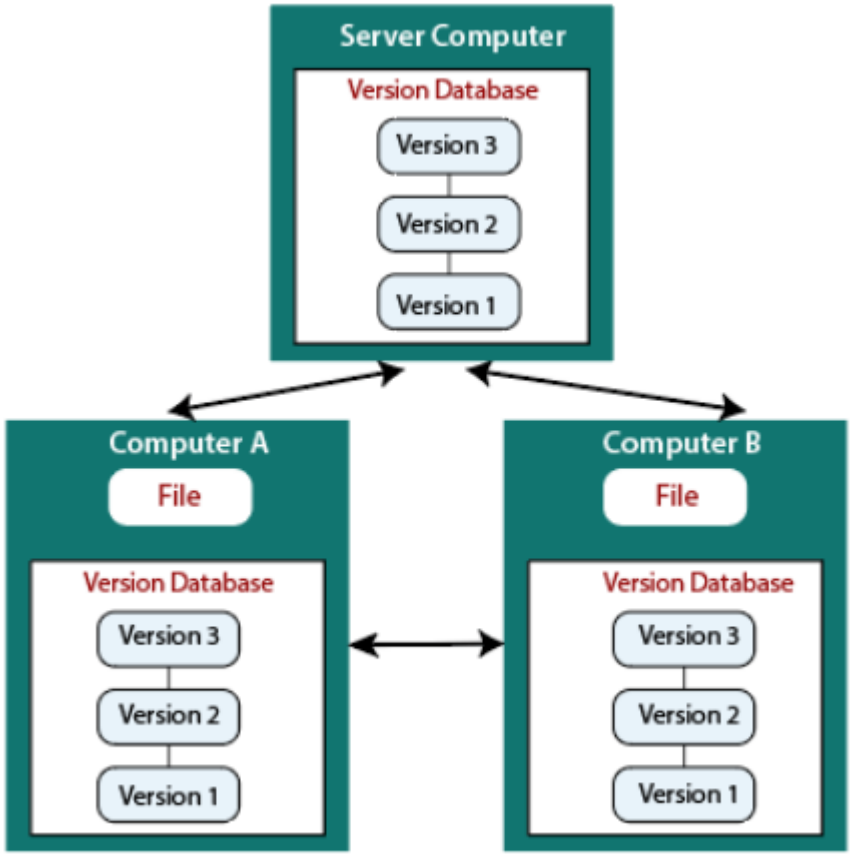
https://www.javatpoint.com/git-version-control-system

Commands:

- **git add –A**
  - Adds your changes to the **index/staging area** (this is a "rough draft space."
- **git commit –m "message explaining your changes"**
  - Adds a "snapshot of your project" (with changes) to the local branch along with your message.
- **git push origin master**
  - Pushes/overwrites these changes you made back to the REMOTE repository.

What is Git?

- A Distributed Version Control System



https://www.javatpoint.com/git-version-control-system