

Lecture 1: Setting our environment and parsing text files

COSC 526: Introduction to Data Mining
Spring 2020



THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Instructors:

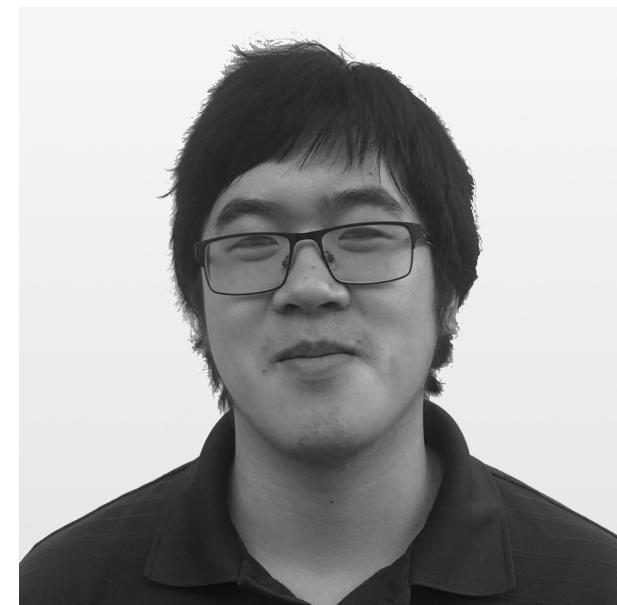


Michela Taufer



Danny Rorabaugh

GRA:



Nigel Tan

Set up our working environments

- We use **git & github** to distribute & collect assignments as well as other class materials (e.g., slides, code, and datasets)
 - Create a GitHub username (if you have not one)
 - Install git and github desktop on your laptop
- We use **Jupyter notebook** for our assignments and project
 - Follow the steps in the file StartHere to install Jupyter
- We use **XSEDE Jetstream** as our platform for assignments and the semester project
 - Introduce its use in the 2nd lecture
 - Create accounts and provide you with access to the cloud in the 4th lecture



Git and GitHub



GitHub and Git

- **GitHub Account:** web-based hosting service for **version control**
 - Use it to distribute and collect assignments, to share class materials (e.g., slides, codes, and datasets)
 - Provide us with your GitHub username
- **Git:** software used by GitHub
 - Install git on your laptop
- **Class GitHub Repository:**
 - Clone the course repository
<https://github.com/CISC879-BigData/courses-UTK-COSC526-S20>



Git and GitHub

- Git: a free, open source distributed version control system
- GitHub: cloud hosted git repositories

Material built from:

<http://rogerdudler.github.io/git-guide/>

<https://marklodato.github.io/visual-git-guide/index-en.html>

https://kbroman.org/github_tutorial/pages/init.html

https://kbroman.org/github_tutorial/pages/routine.html



Install git on your laptop (if you have not yet)

- Windows or Mac
 - Download the GitHub Desktop application:
<https://desktop.github.com>
 - Find GitHub's installation help here:
<https://help.github.com/desktop/guides/getting-started-with-github-desktop/>
- Linux or Mac
 - Install the command-line interface (CLI) of git
 - Find help here:
<https://www.atlassian.com/git/tutorials/install-git>
- Log into github



Connect new repos to github

- Go to [github](#)
- Log in to your account

You don't have a github account?



Connect new repos to github

- Go to [github](#)
- Log in to your account

You don't have a github account?

- Get a [github](#) account
 - Go to the GitHub Sign Up page <https://github.com/join>
 - Create a free account



Connect new repos to github

- Go to [github](#)
- Log in to your account

You don't have a github account?

- Get a [github](#) account
 - Go to the GitHub Sign Up page <https://github.com/join>
 - Create a free account
- Share your github account with us:
 - Complete form <https://forms.gle/CKugke8Dzqjm9tQ89>



Command-line interface (CLI)



CLI: Your first time with git and github (I)

After installing git and github desktop, if you are using CLI:

- Set up git with your user name and email

```
$ git config --global user.name "Your name here"  
$ git config --global user.email "your_email@example.com"
```

- Set up ssh on your computer
 - Look to see if you have files `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.
 - If not, create such public/private keys:

```
$ ssh-keygen -t rsa -C "your_email@example.com"
```



CLI: Your first time with git and github (II)

- Copy your public key (the contents of the newly created `id_rsa.pub` file) into your clipboard – e.g., on Mac

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

- Paste your ssh public key into your github account settings
 - Go to your github [Account Settings](#)
 - Click “[SSH Keys](#)” on the left.
 - Click “Add SSH Key” on the right.
 - Add a label (like “My laptop”) and paste the public key into the big text box.



CLI: Your first time with git and github (III)

- In a terminal/shell, type the following to test it:

```
$ ssh -T git@github.com
```

- If it says something like the following, it worked:

```
Hi username! You've successfully authenticated, but Github does  
not provide shell access.
```



Submit your github username

- Complete form <https://forms.gle/CKugke8Dzqjm9tQ89>



Use git and github

- The routine use of git involves just a few commands:
 - init
 - add and commit
 - push and pull
 - status
 - diff
- You can deal with git and github via:
 - GitHub Desktop (GUI)
 - command line (CLI)



A new repo from scratch

- Create a directory to contain the project
 \$ mkdir YourRepos
- Go into the new directory
- Type *git init*
 \$ *git init*
- Create a new file – ReadMe.md is a good start
 \$ echo "# YourRepos" >> ReadMe.md
- Type *git add* to add the file
 \$ *git add* README.md
- Type *git commit* to commit the file
 \$ git commit -m "first commit"

At this point, your repos is ONLY local



Keep the repo clean

- Create a **.gitignore** file to indicate all of the files you don't want to track in a (sub)directory

```
$ git add .gitignore
```

- Create a **.gitignore_global** file to indicate all of the files you don't want to track for the entire directory. You have to tell git about the global .gitignore

```
$ git add .gitignore_global
```

```
$ git config --global core.excludesfile ~/.gitignore_global
```



.gitignore_global and .gitignore

*~

.*~

.DS_Store

.Rhistory

.RData



Connect new repo to github

- Go to [github](#)
- Log in to your account
- Click the [new repository](#) button in the top-right
- Click the “Create repository” button
- Back to your terminal:
`$ git remote add origin https://github.com/YourAccount/YourRepos.git
$ git push -u origin master`



Add and commit (I)

- Add completely new files as well as to “add” modifications to files that already exist in the repository (e.g., README.md)

```
$ mkdir Lecture01
```

```
$ touch Lecture01/README.md
```

```
$ emacs README.md
```

```
$ git add Lecture01 Lecture01/README.md README.md
```

- Use git commit to add the modifications to the repo

```
$ git commit OR $ git commit -m "Message: 40 / 60 characters"
```

```
$ git commit -m "Fix such and such"
```

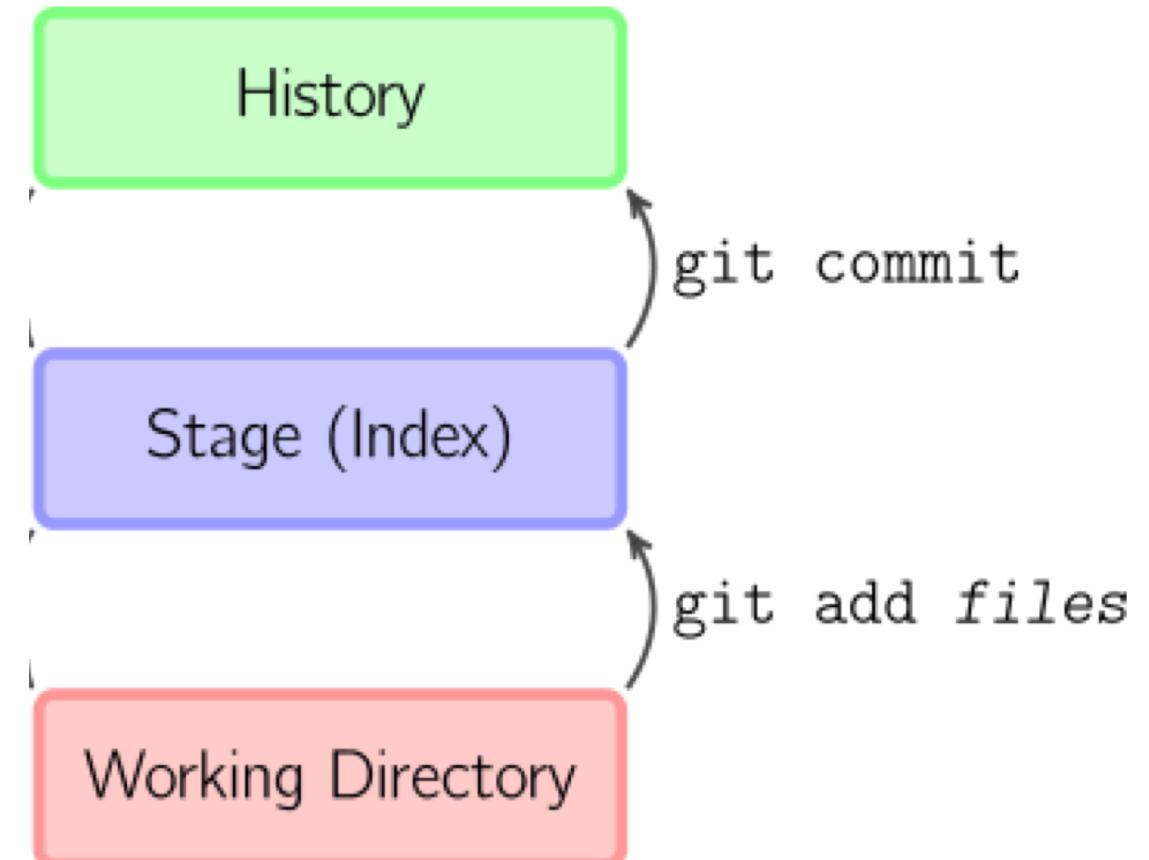


Add and commit (II)

```
$ git add <filename>
```

```
$ git add
```

```
$ git commit -m "Commit message"
```



Push to and pull from github

- After any *git add* and *git commit* commands, changes are in the **HEAD** of your **local working copy**
- To push committed changes to your remote repository - github, type:

```
$ git push
```

- To pull committed changes from your remote repository - github, type:

```
$ git pull
```



Differences github

- To extract differences and changes, type:

```
$ git diff
```

- To extract changes in one specific file, type

```
$ git diff README.md
```

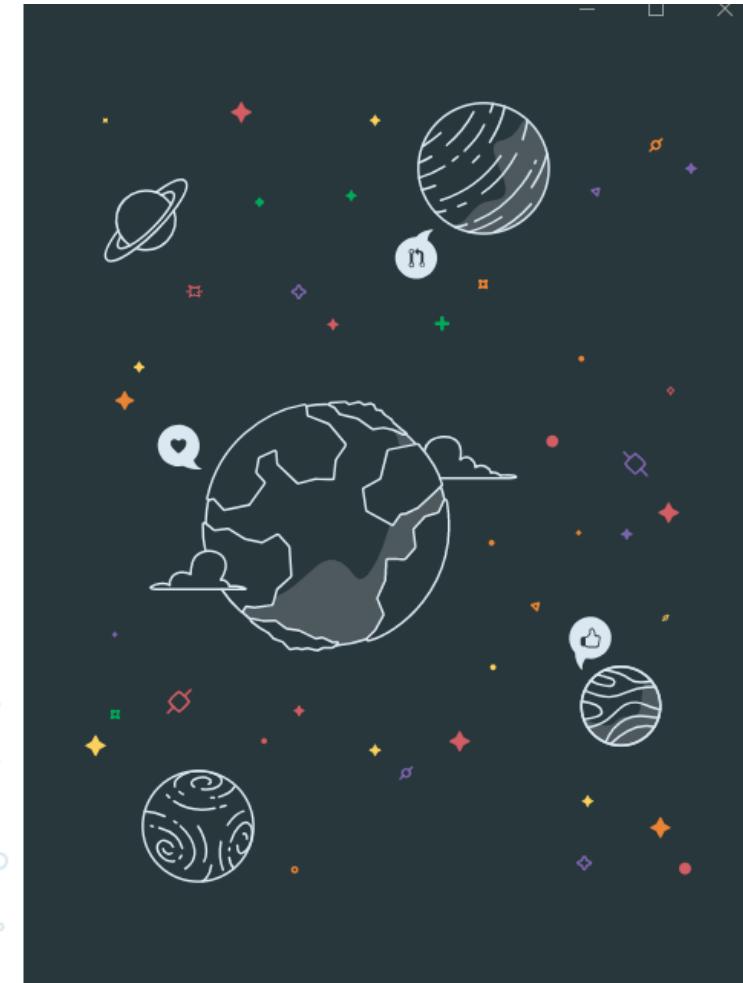
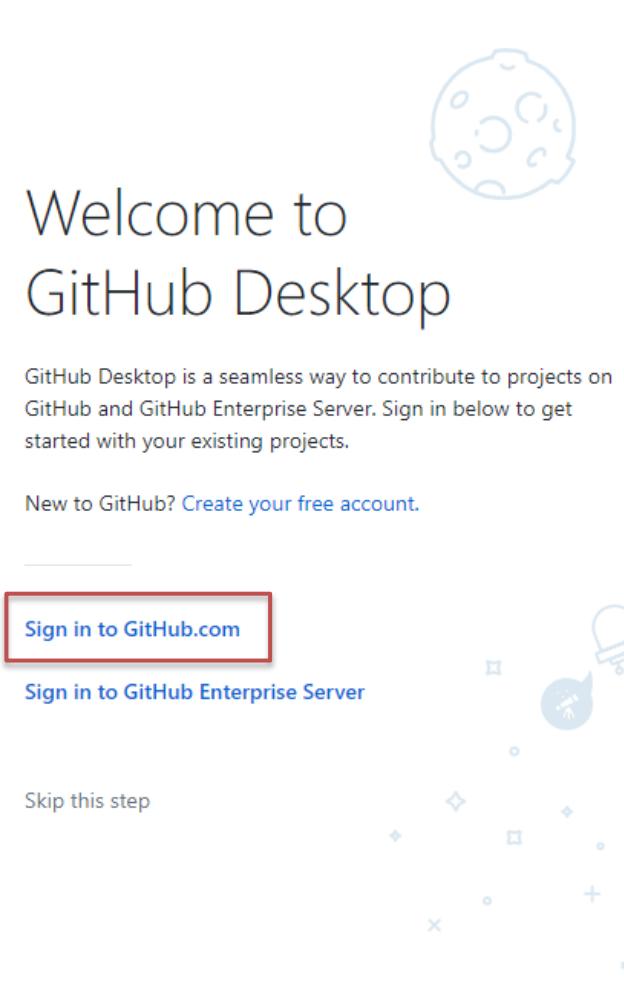


GitHub Desktop application

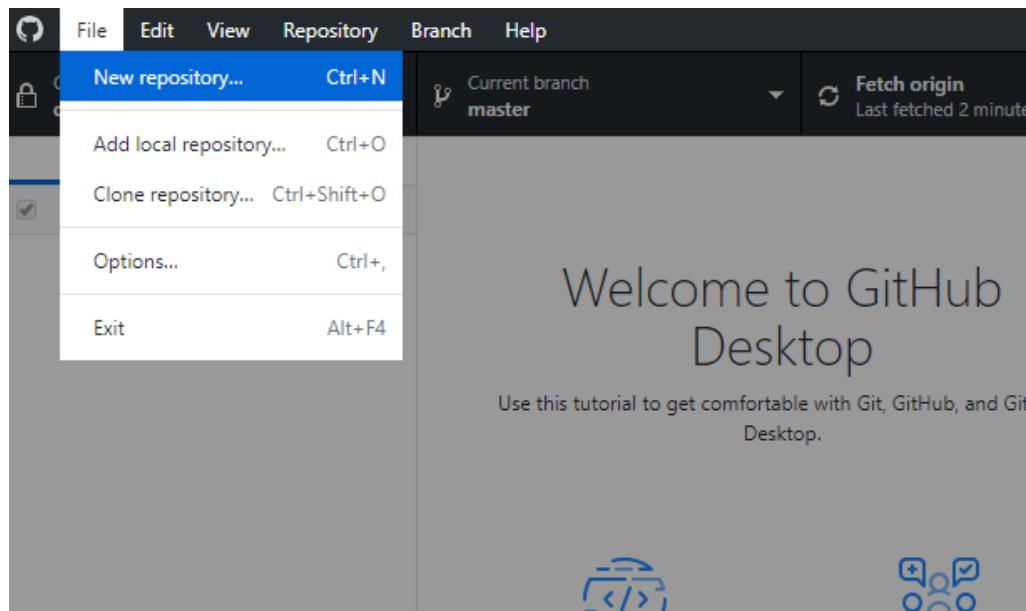


GitHub Desktop: sign-in

- Sign in with your Git username and password
- No need to setup an SSH key!
- If you want the CLI for Git on windows, check out:
<https://gitforwindows.org>



GitHub Desktop: create repo



Create a new repository X

Name

Description

Local path Choose...

Initialize this repository with a README

Git ignore

License

Create repository Cancel

GitHub Desktop: push repo

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Publish your repository to GitHub

This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.

Always available in the toolbar for local repositories or `Ctrl P`

[Publish repository](#)

View the files of your repository in Explorer

Repository menu or `Ctrl Shift F`

[Show in Explorer](#)

Publish repository

[GitHub.com](#)

[GitHub Enterprise Server](#)

Name

example-repo

Description

Keep this code private

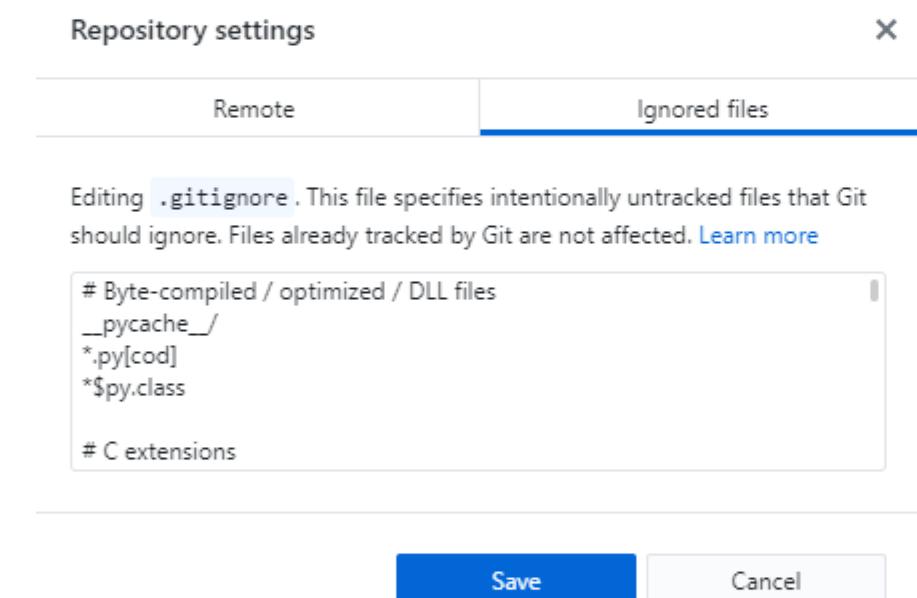
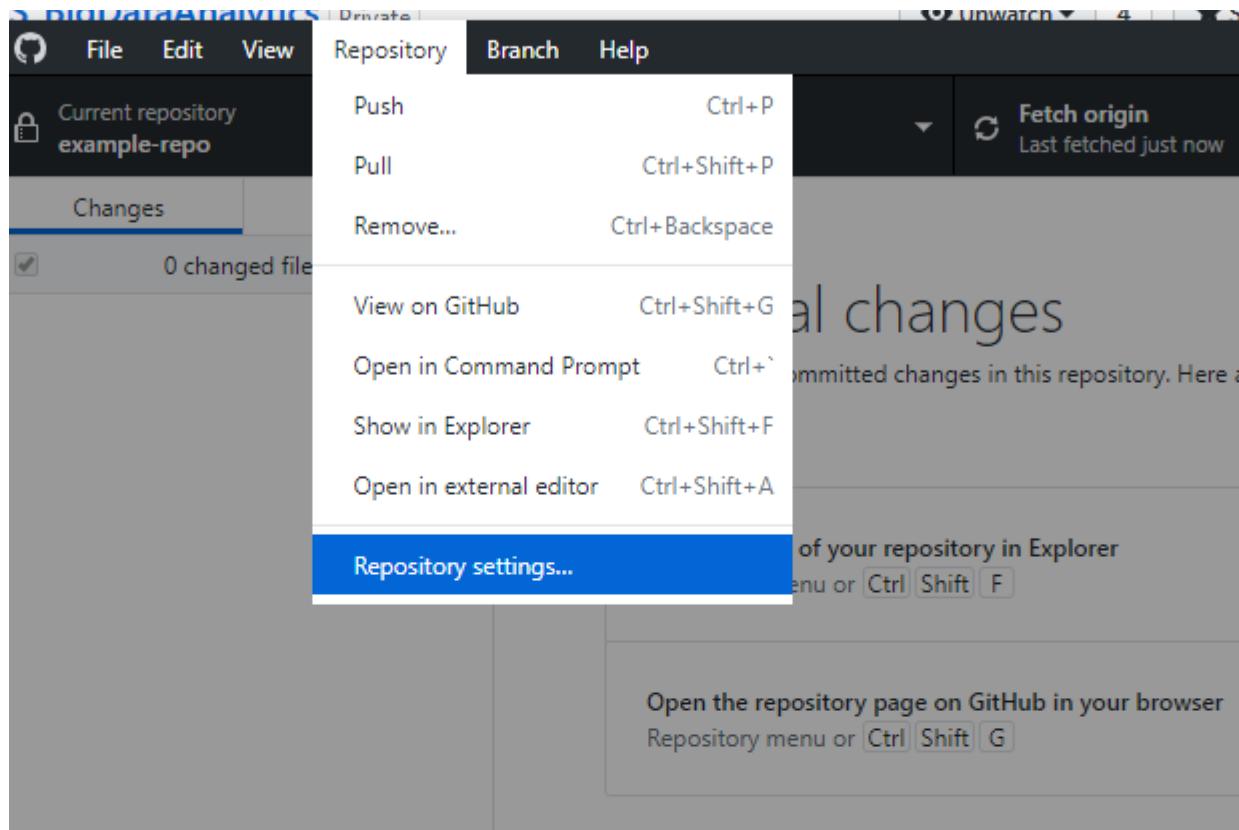
Organization

None

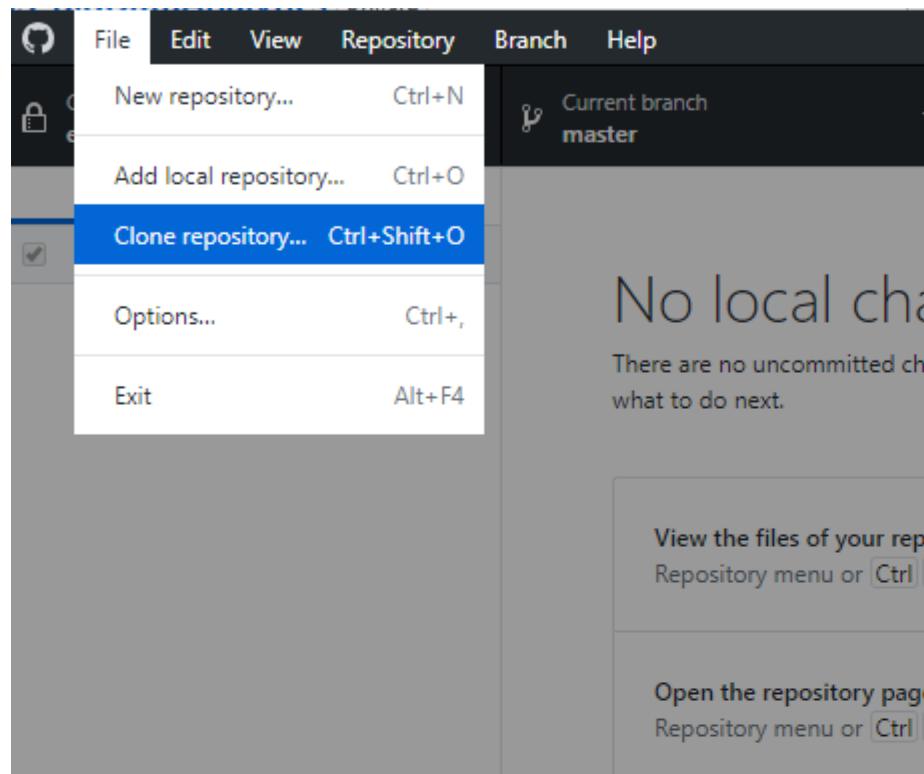
[Publish repository](#)

[Cancel](#)

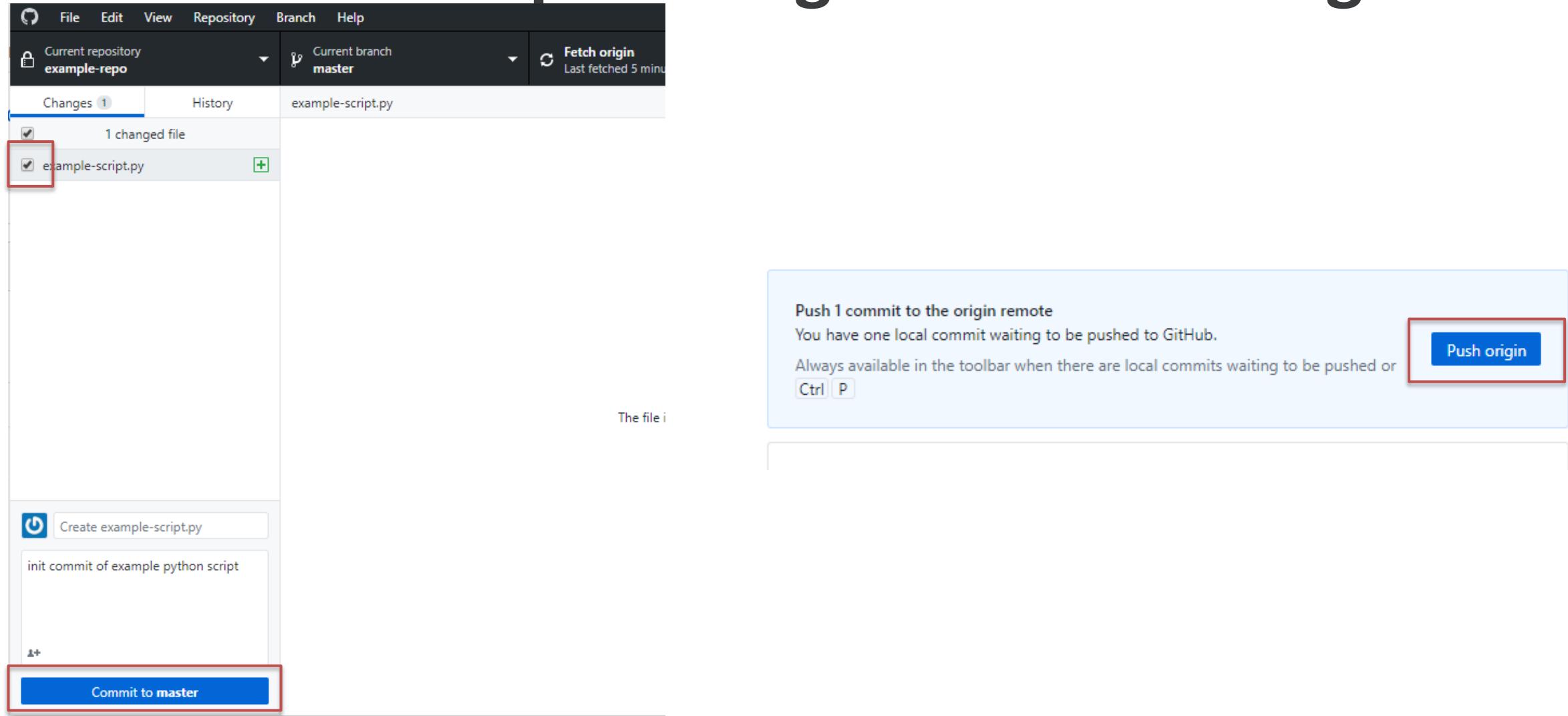
GitHub Desktop: git ignore



GitHub Desktop: clone from git



GitHub Desktop: adding and committing



Python, Anaconda, and Jupyter Notebook

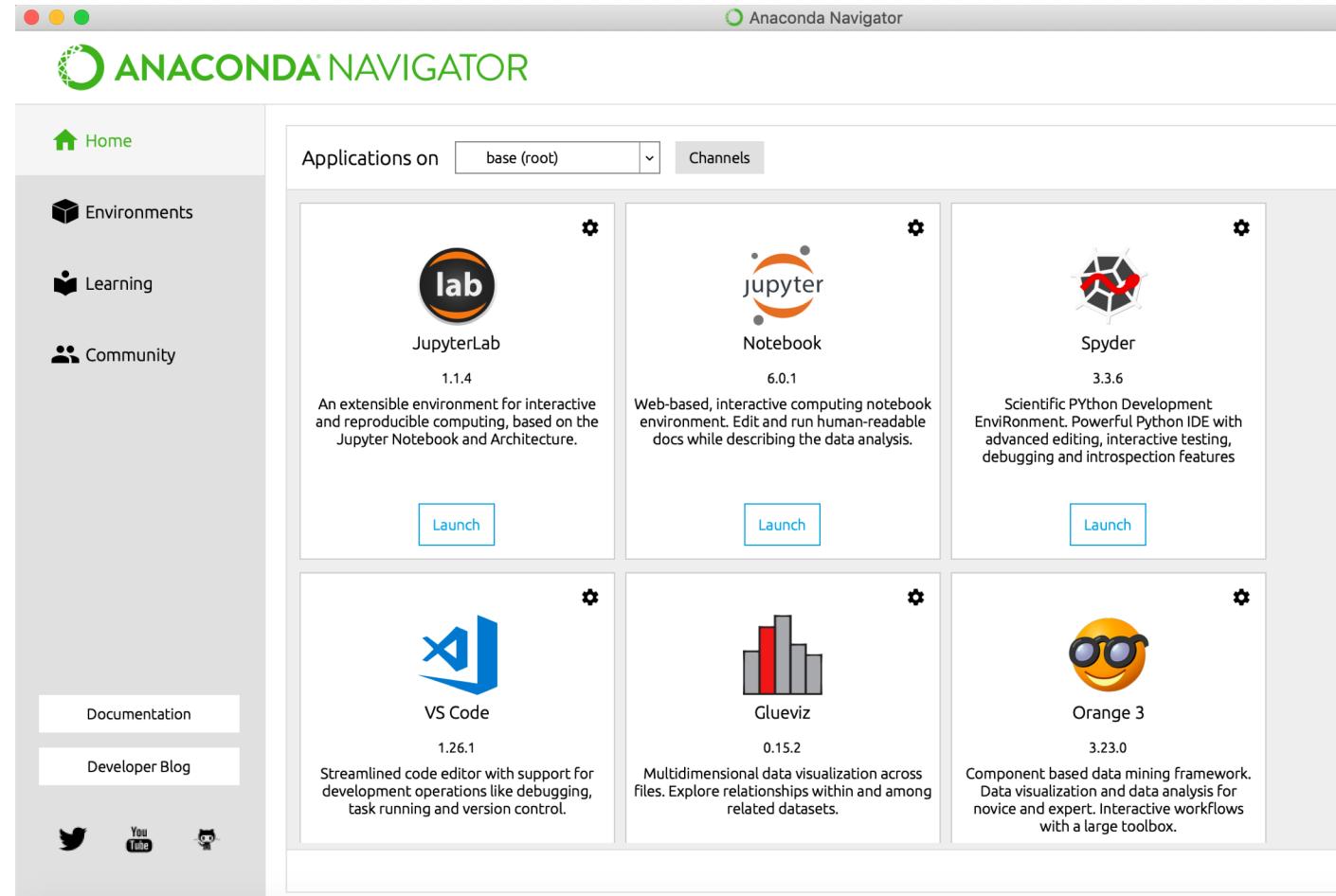


Python and Anaconda

- **Python:** It is Python 3.6!
- **Anaconda:** Python distribution
 - Include many popular packages by default, including Jupyter notebook
 - Make installing additional packages easy
- Your task:
 - Install Anaconda and launch the Jupyter notebook on your laptop by following the steps in the file StartHere.pdf file in course repos (Assignment01)



Anaconda navigator



Jupyter

- **Jupyter:** Our notebook for data analytics
- Programming in a browser
 - Create code in a cell – code in edit mode
 - Run code in a cell – code in command mode
 - Write text before and after code cells – markdown
- Your task:
 - Launch Jupyter from Anaconda GUI (or command line)



Open your assignment directory

The screenshot shows a GitHub repository page for 'CISC879-BigData / 2018F-CS594-CS690'. The repository is private, has 5 pull requests, 0 issues, 0 projects, and 0 wiki pages. The 'Code' tab is selected. The branch is 'master'. The repository path is '2018F-CS594-CS690 / Assignment01 /'. The commit history shows the following entries:

File / Commit Message	Author	Date
..	imnasnainaec StartHere edits.	Latest commit 7873682 3 days ago
images	StartHere edits.	3 days ago
Assignment01.ipynb	Initial commit	4 days ago
StartHere.ipynb	StartHere edits.	3 days ago
StartHere.pdf	StartHere edits.	3 days ago
data.csv	Initial commit	4 days ago
data.tsv	Initial commit	4 days ago

Open your assignment directory

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New ▾

<input type="checkbox"/> 0		/ 00_git_repos / 2018F-CS594-CS690 / Assignment01	Name	Last Modified	File size
<input type="checkbox"/>		..		seconds ago	
<input type="checkbox"/>		images		3 hours ago	
<input type="checkbox"/>		Assignment01.ipynb	Running	3 hours ago	11.1 kB
<input type="checkbox"/>		StartHere.ipynb	Running	an hour ago	6.87 kB
<input type="checkbox"/>		data.csv		3 hours ago	1.06 kB
<input type="checkbox"/>		data.tsv		3 hours ago	1.05 kB
<input type="checkbox"/>		StartHere.pdf		3 hours ago	289 kB

Create your own notebook

jupyter

Files Running Clusters

Select items to perform actions on them.

0 / 00_git_repos / 2018F-CS594-CS690 / Assignment01

Name ↴

..

images

Assignment01.ipynb

StartHere.ipynb

data.csv

data.tsv

StartHere.pdf

Upload New ▾ ⚡

Notebook: Python 3

Other: Text File

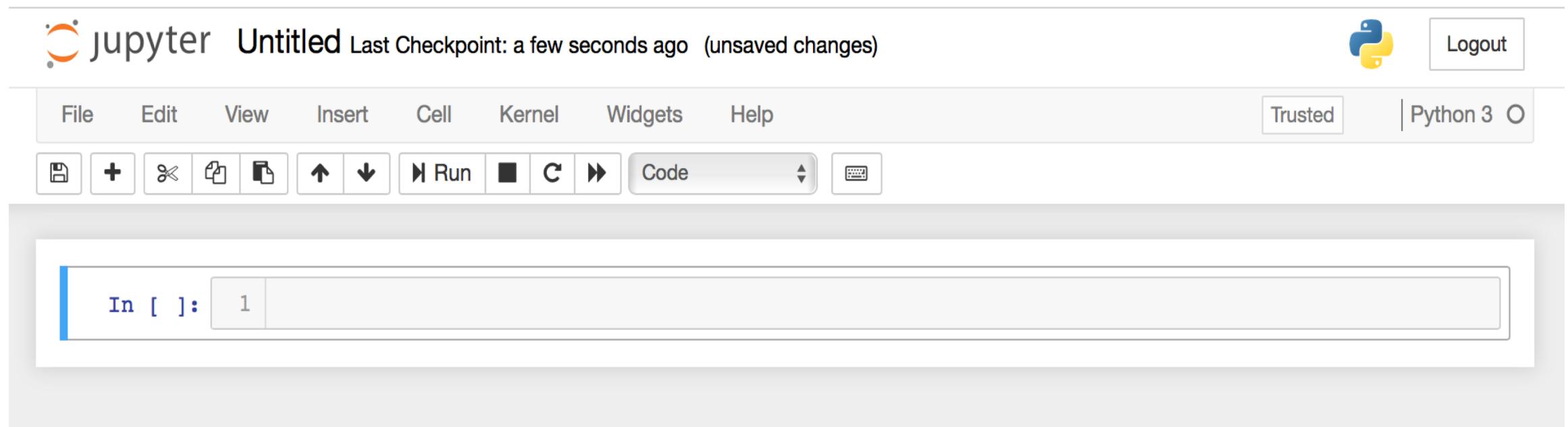
Folder

Terminal

Run	Size
3 hours ago	1.06 kB
3 hours ago	1.05 kB
3 hours ago	289 kB

This screenshot shows the Jupyter Notebook interface. At the top, there's a navigation bar with 'jupyter' logo, 'Files', 'Running', and 'Clusters' tabs. Below the navigation bar, a message says 'Select items to perform actions on them.' A sidebar on the left shows a file tree: '00_git_repos / 2018F-CS594-CS690 / Assignment01'. Inside 'Assignment01', there are folders for '..', 'images', and files for 'Assignment01.ipynb', 'StartHere.ipynb', 'data.csv', 'data.tsv', and 'StartHere.pdf'. On the right, there's a context menu with 'Upload', 'New ▾', and a refresh icon. A dropdown menu under 'New ▾' shows options for creating a new notebook ('Notebook: Python 3') or other types ('Text File', 'Folder', 'Terminal'). Below the menu, a table lists the files in the current directory with their last modified time and size.

Create your own notebook



Create your own notebook

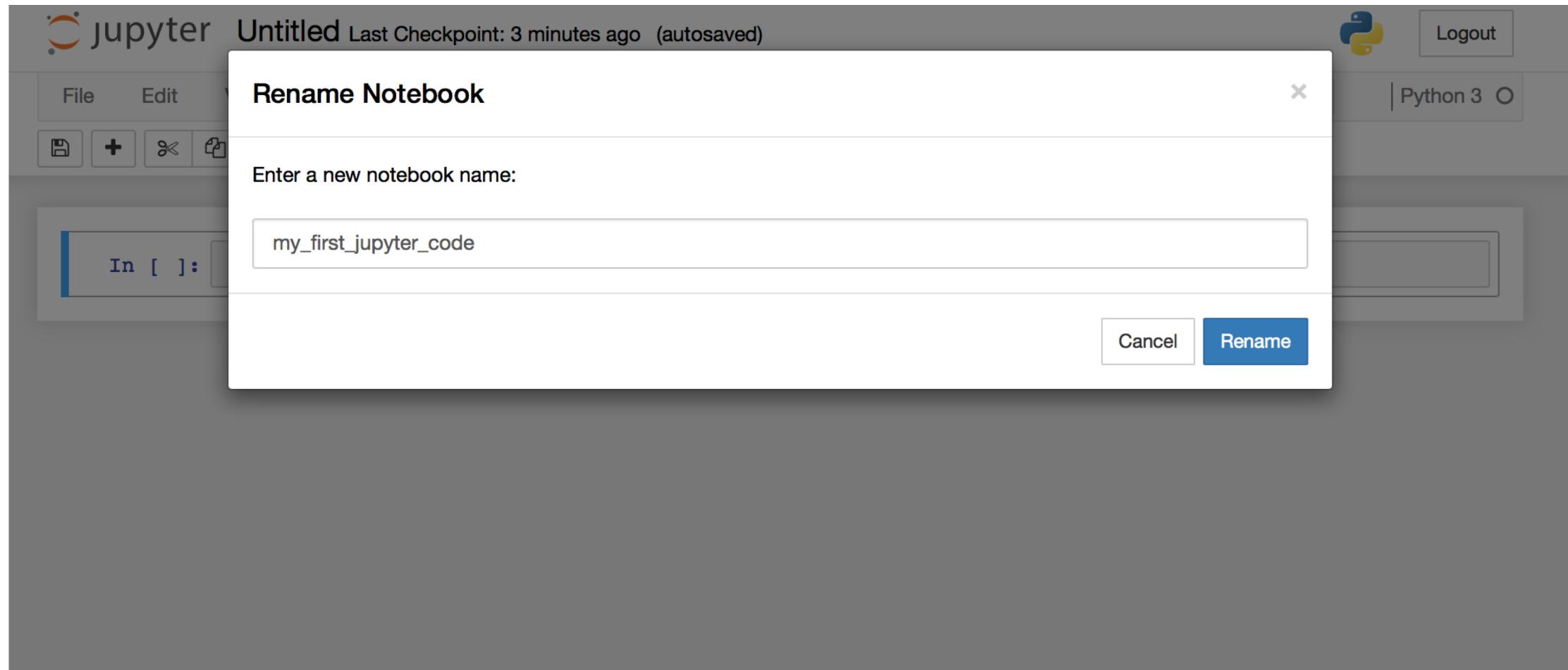
jupyter Untitled Last Checkpoint: a minute ago (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

New Notebook ▾   Run    Code 

Open...
Make a Copy...
Rename...
Save and Checkpoint
Revert to Checkpoint ▾
Print Preview
Download as ▾
Trusted Notebook
Close and Halt

Rename the file

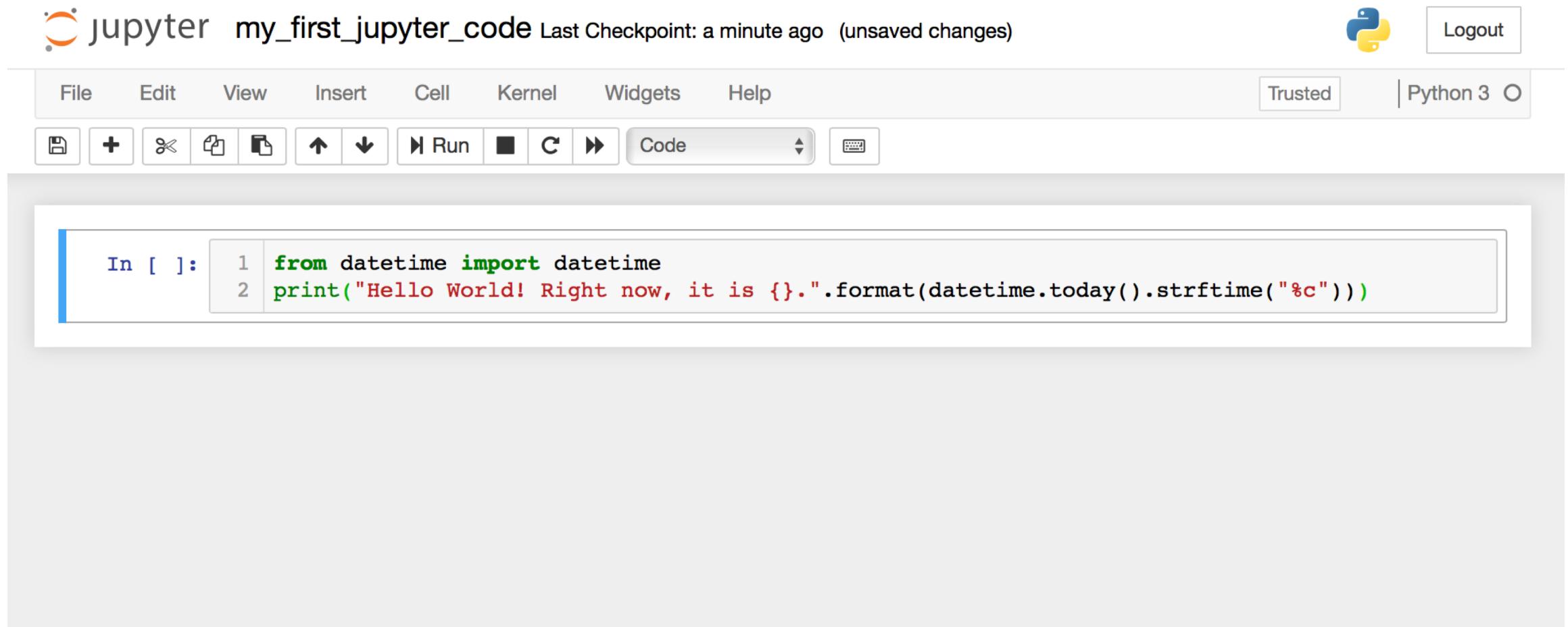


Create your own notebook

The screenshot shows a Jupyter Notebook interface. At the top left is the Jupyter logo. On the right are 'Quit' and 'Logout' buttons. Below the logo is a navigation bar with 'Files' (selected), 'Running', and 'Clusters' tabs. A message 'Select items to perform actions on them.' is displayed above a file list. The file list shows the contents of a directory: '00_git_repos / 2018F-CS594-CS690 / Assignment01'. The files listed are:

	Name	Last Modified	File size
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	images	3 hours ago	
<input type="checkbox"/>	Assignment01.ipynb	Running 3 hours ago	11.1 kB
<input type="checkbox"/>	my_first_jupyter_code.ipynb	Running seconds ago	555 B
<input type="checkbox"/>	StartHere.ipynb	Running an hour ago	6.87 kB
<input type="checkbox"/>	data.csv	3 hours ago	1.06 kB
<input type="checkbox"/>	data.tsv	3 hours ago	1.05 kB
<input type="checkbox"/>	StartHere.pdf	3 hours ago	289 kB

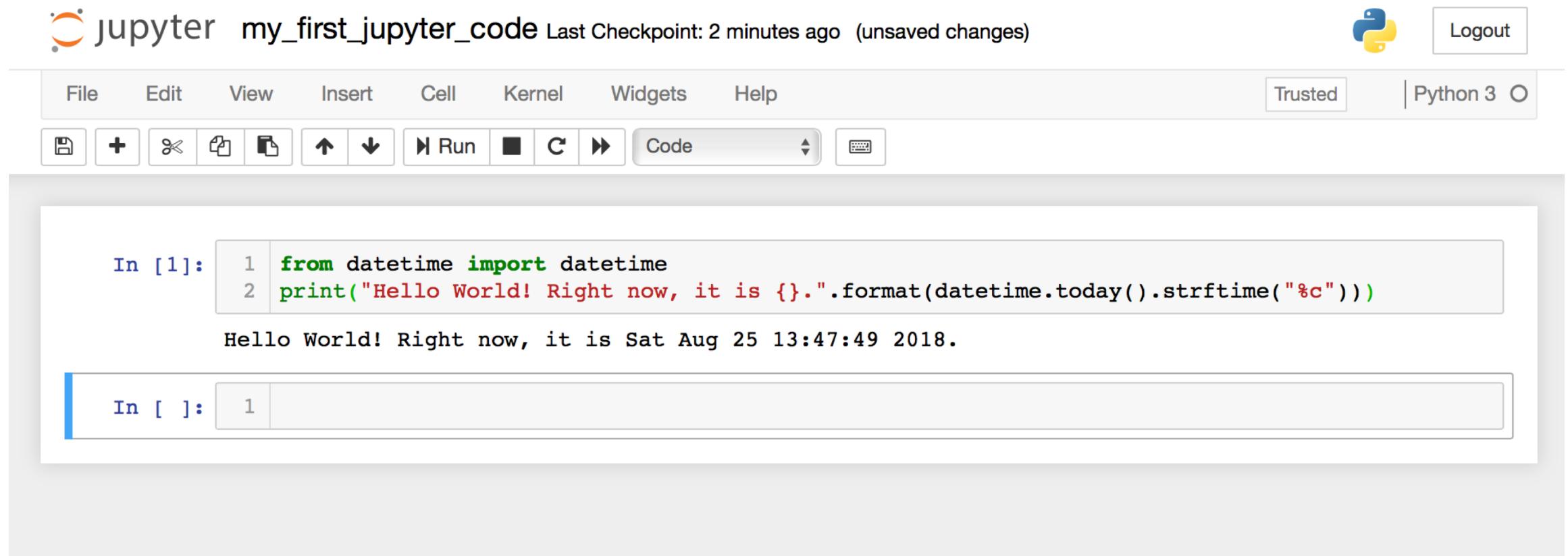
Create a code cell



The screenshot shows a Jupyter Notebook interface. At the top, there's a header bar with the Jupyter logo, the title "jupyter my_first_jupyter_code", a timestamp "Last Checkpoint: a minute ago (unsaved changes)", a Python logo icon, and a "Logout" button. Below the header is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu are "Trusted" and "Python 3" status indicators. A toolbar below the menu contains icons for file operations like saving, creating, and deleting, along with buttons for "Run", "Cell", "Code", and "Kernel". The main area is a code cell with a blue header bar labeled "In []:". Inside the cell, two lines of Python code are displayed:

```
In [ ]: 1 from datetime import datetime  
2 print("Hello World! Right now, it is {}".format(datetime.today().strftime("%c")))
```

Run your code



The screenshot shows a Jupyter Notebook interface. At the top, there's a header with the logo, the notebook name "my_first_jupyter_code", the last checkpoint time ("2 minutes ago"), and status ("unsaved changes"). To the right are a Python logo icon, a "Logout" button, and a "Trusted" badge. Below the header is a toolbar with various icons for file operations like saving, opening, and creating new files, as well as cell controls like running and kernel restarts.

The main area contains a code cell labeled "In [1]". The cell contains two lines of Python code:

```
In [1]: 1 from datetime import datetime  
        2 print("Hello World! Right now, it is {}".format(datetime.today().strftime("%c")))
```

The output of the cell is displayed below it:

```
Hello World! Right now, it is Sat Aug 25 13:47:49 2018.
```

Below the cell, there's another input field labeled "In []:" with the number "1" inside, indicating the next cell to be run.

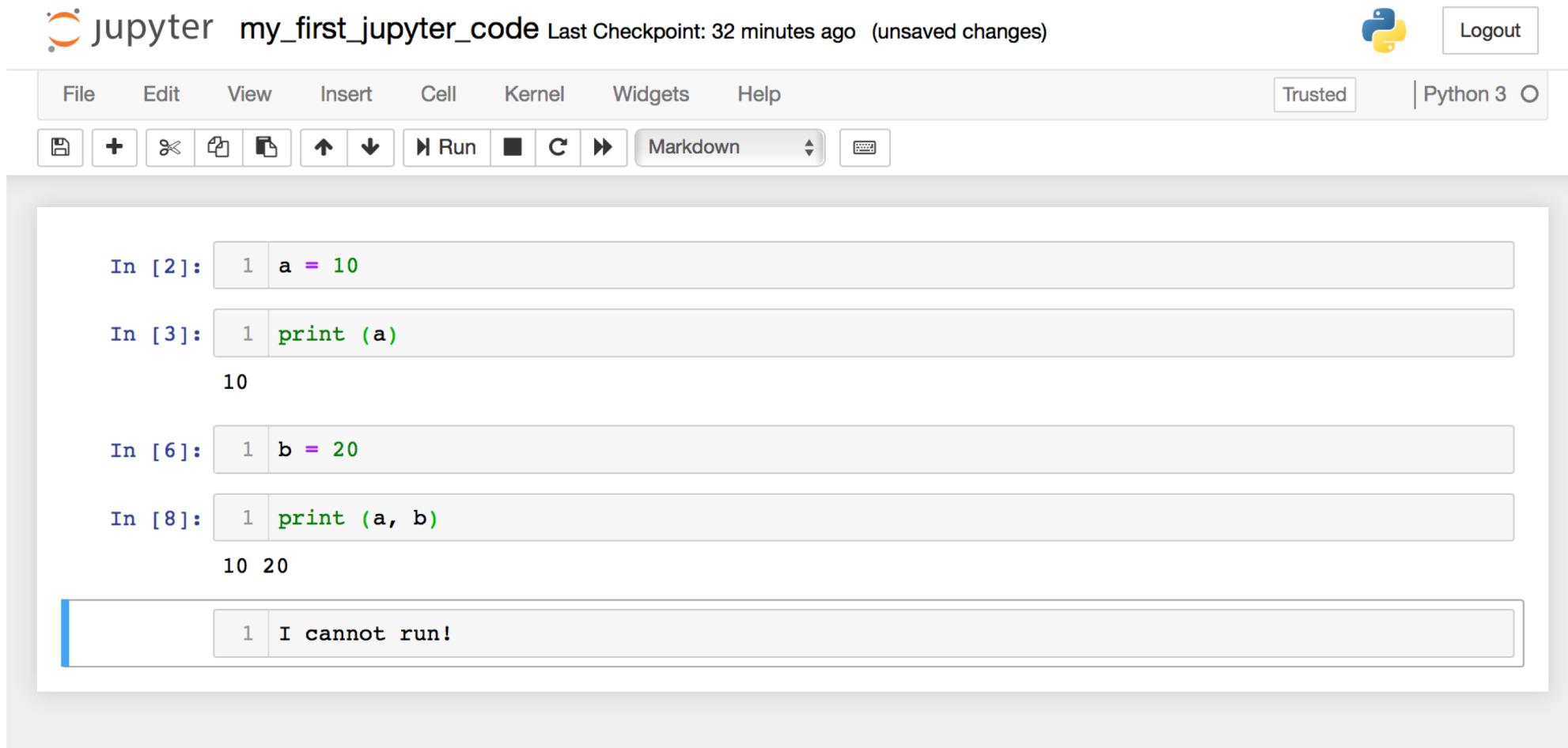


Propagations: from cell to cell

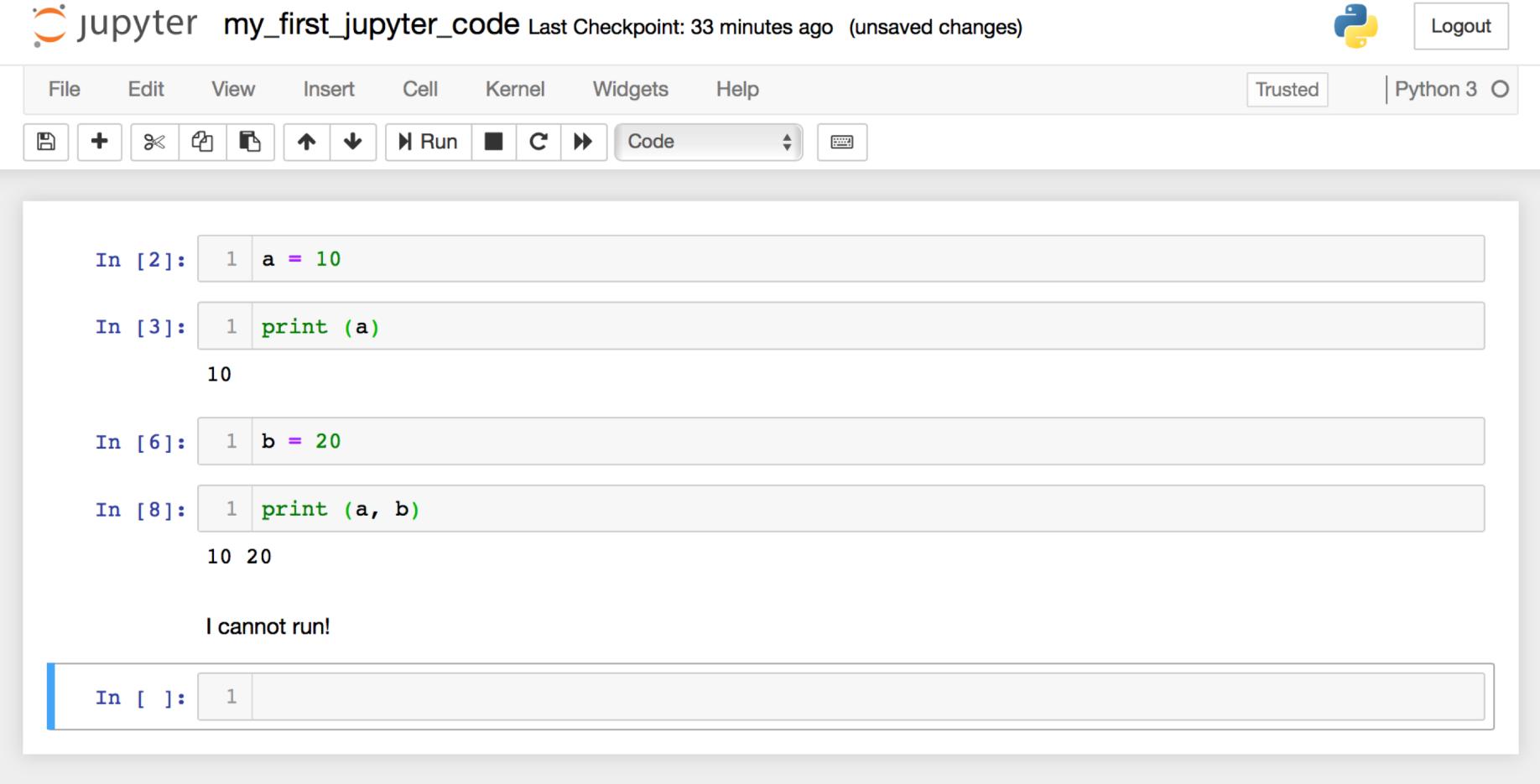
The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** jupyter my_first_jupyter_code Last Checkpoint: 28 minutes ago (autosaved) Python 3
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Notebook saved, Trusted.
- Code Cells:**
 - In [2]: `1 a = 10`
 - In [3]: `1 print (a)`
10
 - In [6]: `1 b = 20`
 - In [7]: `1 print (a, b)`
10 20

Add text to your notebook



Add text to your notebook



The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** jupyter my_first_jupyter_code Last Checkpoint: 33 minutes ago (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3
- Cells:**
 - In [2]: `1 a = 10`
 - In [3]: `1 print (a)`
10
 - In [6]: `1 b = 20`
 - In [8]: `1 print (a, b)`
10 20
 - In []: `1` (The cell is currently active, indicated by a blue border.)
- Message:** I cannot run!

Course Repository



Clone the course repository

<https://github.com/CISC879-BigData/courses-UTK-COSC526-S20>

CISC879-BigData / courses-UTK-COSC526-S20 Private

Unwatch 5 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Course repos Spring 2020 semester - COSC 526 course - material distributed to students attending the course Edit

Manage topics

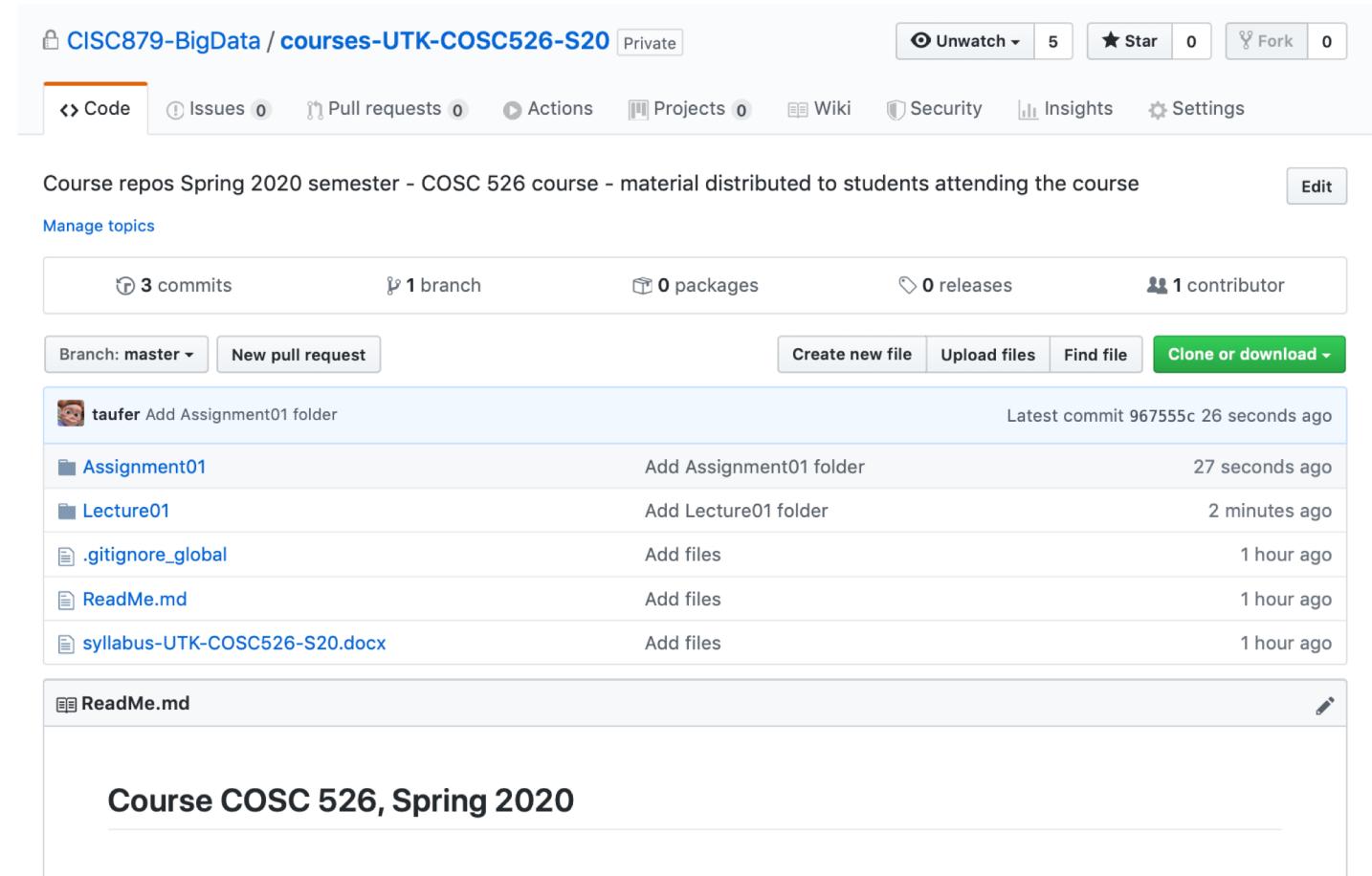
3 commits 1 branch 0 packages 0 releases 1 contributor

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

taufer	Add Assignment01 folder	Latest commit 967555c 26 seconds ago
Assignment01	Add Assignment01 folder	27 seconds ago
Lecture01	Add Lecture01 folder	2 minutes ago
.gitignore_global	Add files	1 hour ago
ReadMe.md	Add files	1 hour ago
syllabus-UTK-COSC526-S20.docx	Add files	1 hour ago

ReadMe.md

Course COSC 526, Spring 2020



Clone the course repository

[CISC879-BigData / courses-UTK-COSC526-S20](#) Private

Unwatch 5 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Course repos Spring 2020 semester - COSC 526 course - material distributed to students attending the course [Edit](#)

Manage topics

3 commits 1 branch 0 packages 0 releases 1 contributor

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

taufer	Add Assignment01 folder
Assignment01	Add Assignment01 folder
Lecture01	Add Lecture01 folder
.gitignore_global	Add files
ReadMe.md	Add files
syllabus-UTK-COSC526-S20.docx	Add files

Clone with HTTPS [Use SSH](#)
Use Git or checkout with SVN using the web URL.
<https://github.com/CISC879-BigData/courses-UTK-COSC526-S20> [Copy](#)

[Open in Desktop](#) [Download ZIP](#)

1 hour ago

ReadMe.md

Course COSC 526, Spring 2020

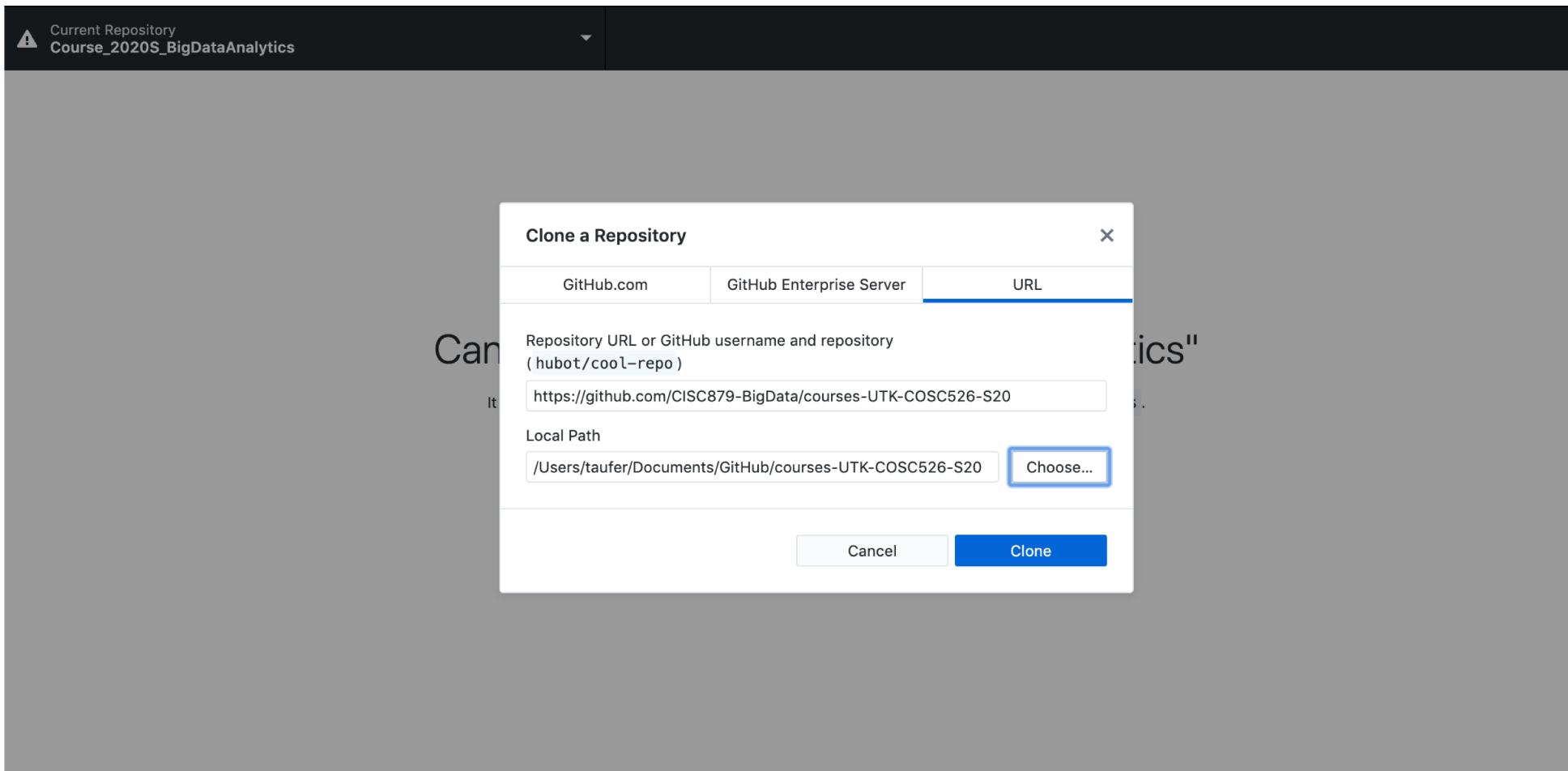


Open from CLI

```
$ git clone https://github.com/CISC879-BigData/courses-UTK-COSC526-S20.git
```



Open from GitHub desktop



Practical programs



Assignment 1: Dealing with text

- Reading in, parsing, and processing delimiter-separated values stored in files – **comma-separated values (csv)** and **tab-separated values (tsv)**
 - Count (and print) the number of rows of data (header is excluded) in the csv file
 - Count (and print) the number of columns of data in the csv file
 - Calculate (and print) the average of the values that are in the "age" column - You can assume each age in the file is an integer, but the average should be calculated as a float



Assignment 1: Dealing with different formats

- Converting the unicode-formatted names into ascii-formatted names
 - Use the provided transliteration dictionary that maps several common unicode characters to their ascii transliteration to convert the unicode strings to ascii



Assignment 1: Practical tasks

- Create your own GitHub account
- Send your GitHub username to
<https://forms.gle/CKugke8Dzqjm9tQ89>
(NOTE THIS IS DUE ON FRIDAY JAN 17 before 8AM ET)
- Install Git on your laptop



Assignment 1: Answer free-form questions

Answer with codes:

- Your solutions for Problems 1 & 2 probably share a lot of code in common.
You might even have copied-and-pasted from Problem 1 into Problem 2.
Refactor *parse_delimited_file* to be useful in both problems.
- Are there any pre-built Python packages that could help you solve these problems? If yes, refactor your solutions to use those packages.

Answer with text:

- Describe the challenges you faced in addressing these tasks and how you overcame these challenges.
- Did you work with other students on this assignment? If yes, how did you help them? How did they help you?



For the next week

- Get your solution done before our next class
 - This week we are not pushing the solution into your own repos yet
- Next week
 - More about private GitHub repos
 - Pull your solution into your GitHub
 - More practice with Jupyter, Python, and problem solving



