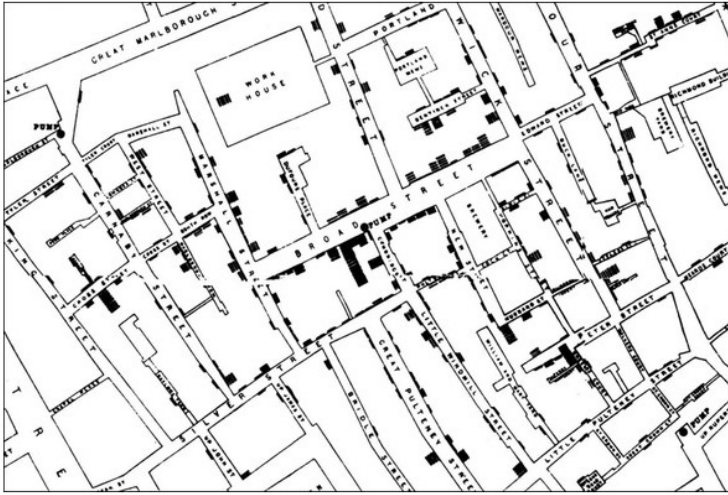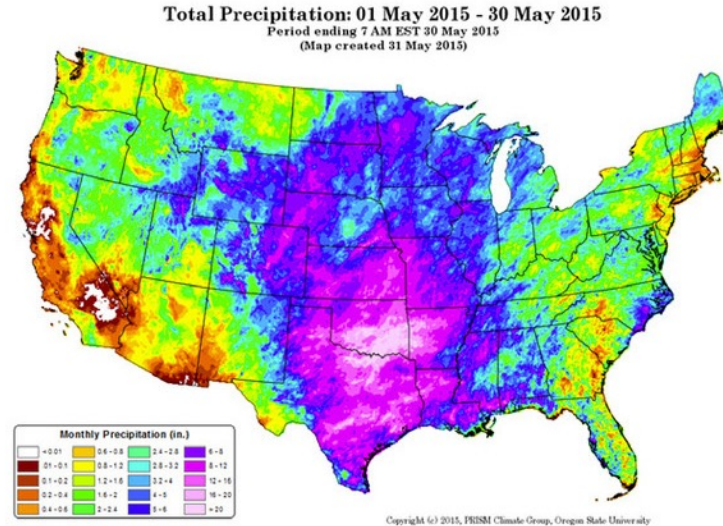# *Introduction to GIS Programming*

Overview

- History of geoprocessing

- Benefits of programming

- What is Python and why popular?

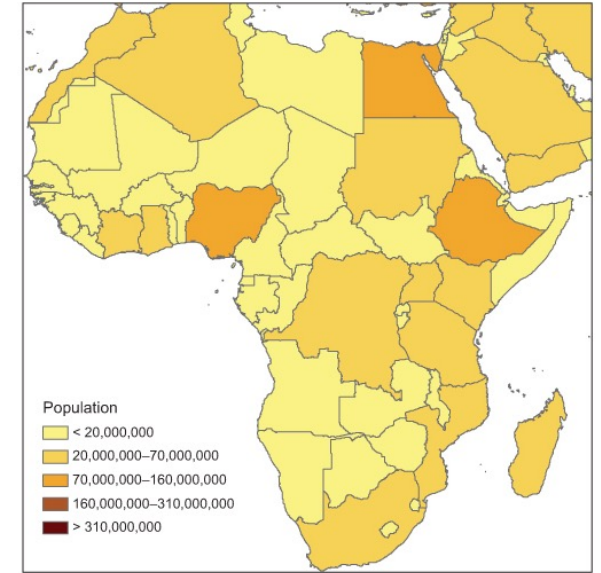- Introduction to Jupyter Notebook & GitHub

- Helpful resources

History of geoprocessing and GIS

John Snow,
Soho cholera outbreak of 1854

Desktop-based
Geographic Information System

Drawbacks of desktop-based GIS

- Expensive; commercial GIS software such as Esri ArcMap and ArcGIS Pro

- OS incompatibility; Esri software does not work on Mac

- Predefined toolsets; not able to modify things in 'black box'

- Tedious mouse clicks

- Hard to save sequences of running tools

Benefits of programming

- Free and open-source languages

- Runs on any operating systems: Windows, Linux, Mac

- Being able to investigate and modify logics inside of the 'black box'

- Automation; no more clicking mouse

- Reproducibility; redo the task easily as long as you have your code

Types of programming (Platform)

- Desktop
  - Things that show up in your start menu
- Web
  - Things you can get to from a web browser
- Scripts
  - Things that you run from your computer but are not in your start menu
- High Performance Computing (HPC)
  - Things that you would run on a supercomputer
- Apps
  - Things that you run on your cell phone

Kinds of programming languages

- **C#**; primarily for Windows

- **VBA**; primarily for Windows

- **C** & **C++**; most powerful fundamental language but hard to learn

- **Java**; works on any platform; was popular (some desktop apps that out of style)

- **HTML**; web development and often combined with **CSS** and **JavaScript**

- **R**; specialized in statistics but also popular in GIS

- **PYTHON**; the focus language of our course

What is Python and why popular?

- History
  - Conceived in late 1980s by Guido van Rossum
  - Python 2 released in 2000 and depreciated in 2020
  - Python 3 announced in 2008 (active)

- Strengths:
  - Intuitive language; pretty easy to learn
  - High-level language, can do a lot with relatively little code
  - Fairly popular among high-level languages
  - Robust support for object-oriented programing
  - Support for integration with other languages (e.g., GDAL binding)

Python dev environment

- Python file (*.py) with code editor (e.g., PyCharm, Atom, Visual Studio Code)
  - Local; only can run an entire python file at a time

- IDLE (Integrated Development and Learning Environment)
  - Local; interactive interface; instant output
  - Run a line at a time

- Jupyter Notebook
  - Web application; internet may be required
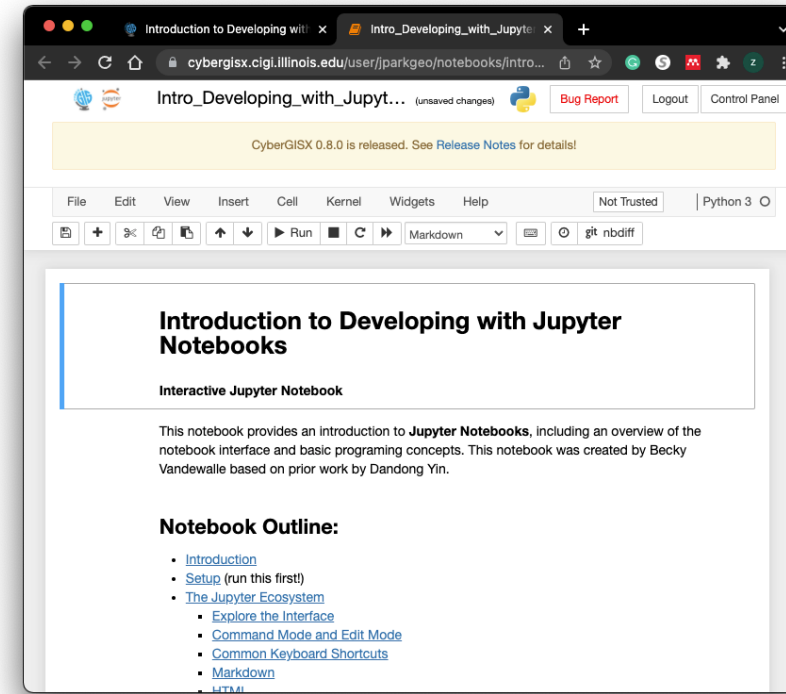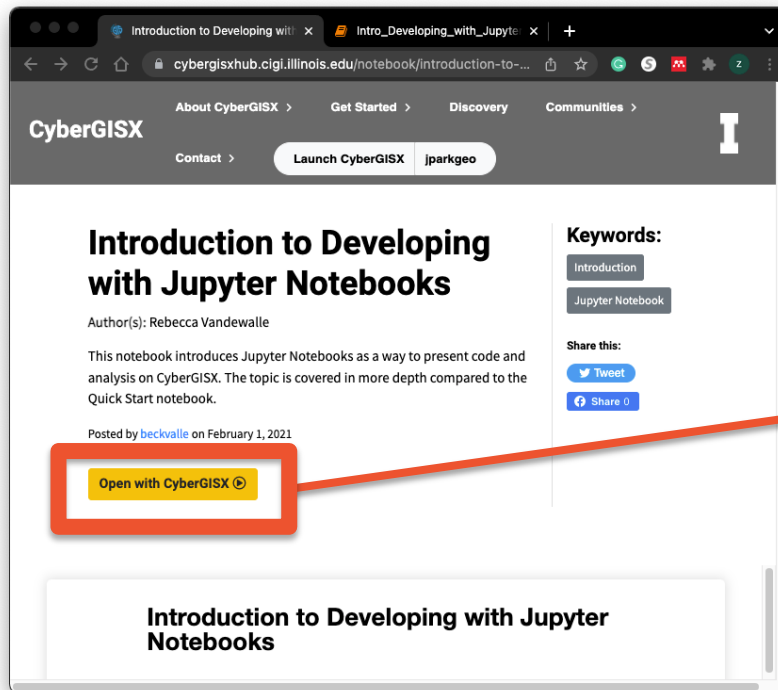 - Can run code partially or entirely as needed
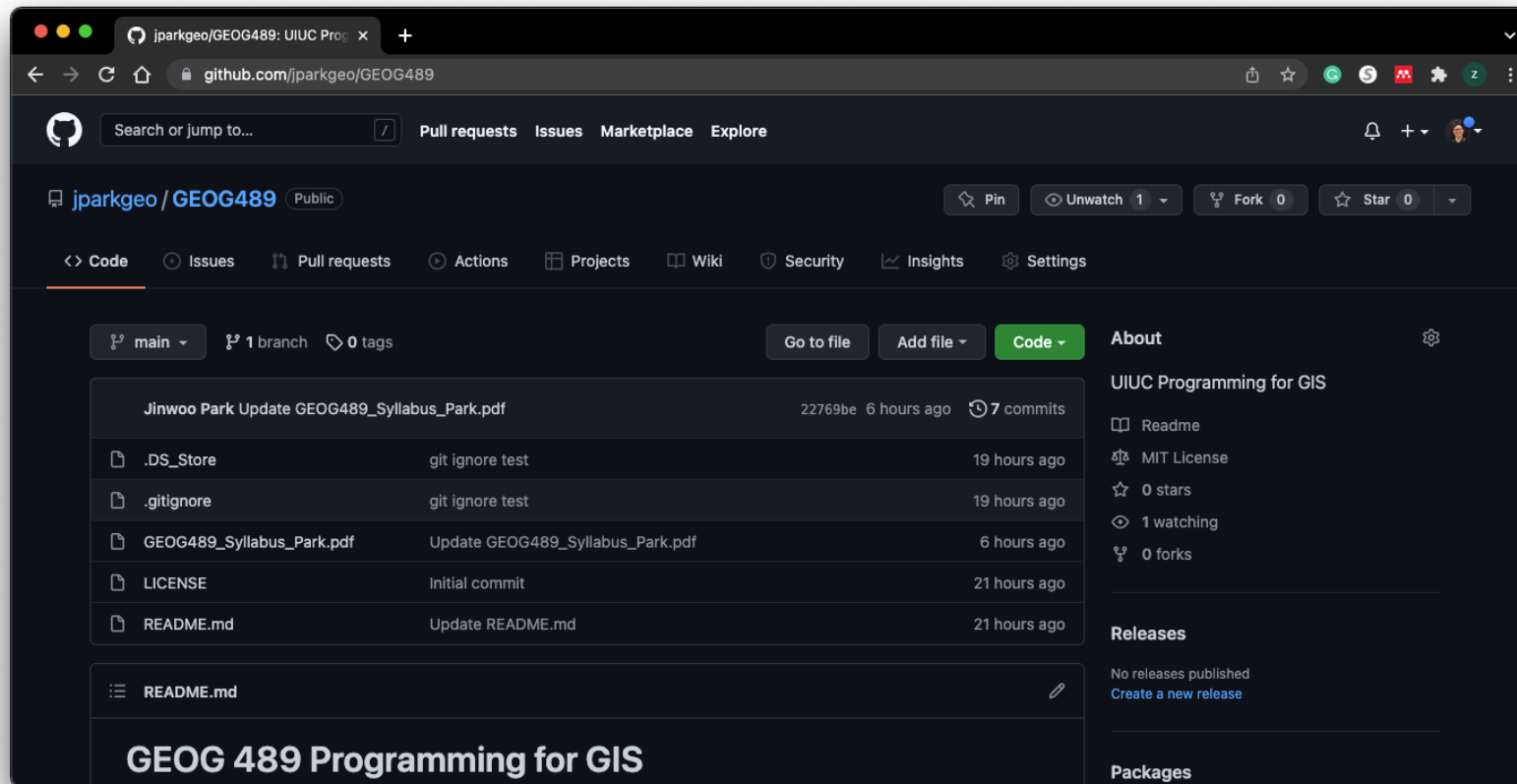
Set up your Jupyter notebook

- https://cybergisxhub.cigi.illinois.edu/get-started/

Intro to Jupyter notebook

- https://cybergisxhub.cigi.illinois.edu/notebook/jupyter-notebooks-quick-start-2/
- https://cybergisxhub.cigi.illinois.edu/notebook/introduction-to-developing-with-jupyter-notebooks/ (optional)

Department of Geography & Geographic Information Science

GitHub (Code repository)

- https://github.com/jparkgeo/GEOG489

Lab 0: Get ready for GIS programming

- The instruction is available at
  [https://github.com/jparkgeo/GEOG489/blob/main/Labs/Lab0/Lab0_Ready_y_for_GIS_Programming.pdf](https://github.com/jparkgeo/GEOG489/blob/main/Labs/Lab0/Lab0_Get_Ready_y_for_GIS_Programming.pdf)

- Tasks
  – How to set up Jupyter notebook on CyberGISX
  – How to navigate GitHub
  – How to interact with a Jupyter Notebook

Department of Geography & Geographic Information Science

Helpful resources

- Official Python Tutorial:
  - https://docs.python.org/3/tutorial/index.html
- Stack Overflow (Q&A Forum):
  - https://stackoverflow.com/
- W3schools:
  - https://www.w3schools.com/python/default.asp
- GeeksforGeeks:
  - https://www.geeksforgeeks.org/python-programming-language/?ref=shm

Next class

- Python basics
  - Data types: Binary, Integer, Float, Boolean, etc.
  - Data containers: List, Dictionary, Tuple, Set
  - Basic functions: Type(), Range()

- Reading (recommended)
  - Geoprocessing with Python, Chapter 2

Q&A

Jinwoo Park
jparkgeo@illinois.edu