

## 01.2\_Installation-Notebook-GitHub

August 23, 2023

### 1 Introduction to Python for Open Source Geocomputation



- Instructor: Dr. Wei Kang

Content:

- [Installation](#)
- [Teaching and Learning with Jupyter](#)
- [GitHub](#)

### 2 Installation

#### 2.1 What is Anaconda?

Anaconda: Anaconda Distribution equips individuals to easily search and install thousands of Python packages and access a vast library of community content and support.

- Anaconda Repository: features over 8,000 open-source data science and machine learning packages, Anaconda-built and compiled for all major operating systems and architectures.
- Conda: is an open-source package and environment management system that runs on Windows, macOS, and Linux.
- Anaconda Navigator (Graphical User Interface (GUI)): lets you easily manage integrated applications, packages, and environments without using the command line.

#### 2.2 Installing Anaconda on your own computer

##### 2.2.1 Step 1

Download [Anaconda Installer](#)

# Free Download

Everything you need to get started in data science on your workstation.

- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

[Code in the Cloud](#)
[Download](#)

Get Additional Installers



## 2.2.2 Step 2

Install Anaconda

- [Windows Anaconda installation instructions](#)
- [macOS Annaconda installation instructions](#)
- [Linux Anaconda installation instructions](#)

Once you have installed Anaconda, you can explore the options for interacting with Python through Anaconda: [Getting started with Anaconda](#).

## 2.2.3 Step 3

Two ways of using Anaconda to manage python packages and environments on your computer:

[Getting Started with Anaconda](#)

### 1. *conda*: command line

- [conda tutorial](#)
- [Anaconda prompt/terminal](#)
- powershell

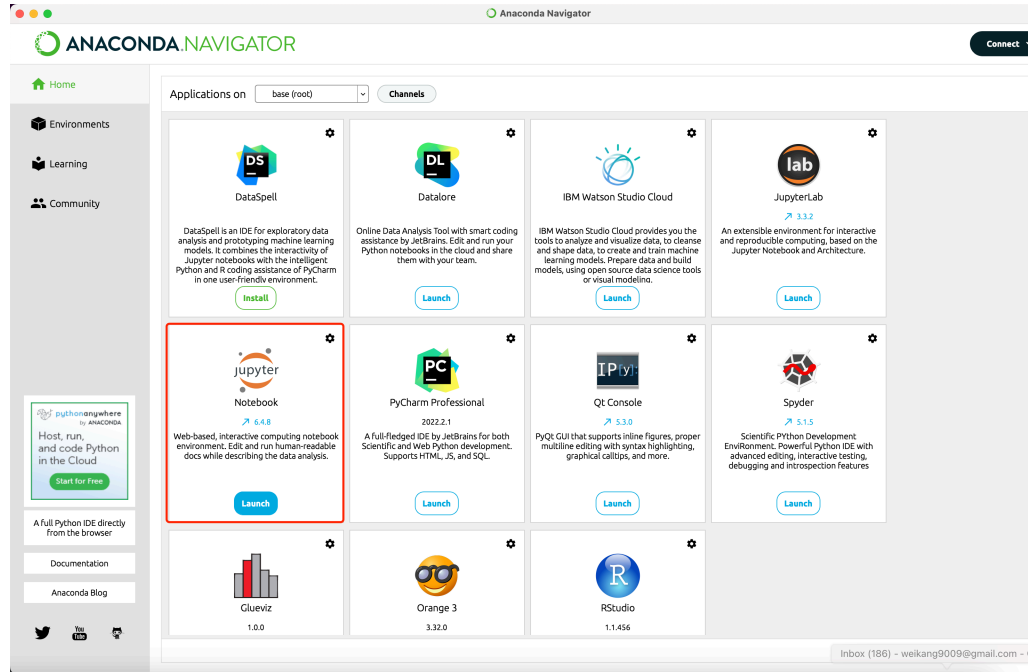
### 2. *Anaconda Navigator*: Graphical User Interface

- [Anaconda Navigator tutorial](#)

## 2.2.4 Step 4

Launching your first Jupyter Notebook

- From an Anaconda Prompt or terminal: type `jupyter notebook` and press Enter.
  - Windows:
    - \* From the Start menu, search for and open “Anaconda Prompt”
    - \* From the Start menu, search for and open “powershell”
  - Mac/Linux: Open a terminal window.



- From *Anaconda Navigator*:

[ ]:

## 3 Teaching and Learning with Jupyter

### 3.1 What is Jupyter Notebook?

Notebooks are documents containing **text narratives** with images and math, combined with **executable code** (many languages are supported) and the **output** of that code.

- interactive computing environment for working with Python (and other languages)
- Similar in spirit to a scientific notebook (but much more)
  - Live code
  - Interactive widgets
  - Plots
  - Narrative text
  - Equations
  - Images
  - Video
- multiple sharing mechanisms
  - GitHub
  - binder
  - nbviewer
  - Jupyter Book

- other traditional format (.pdf)
- literate programming
- simple JSON format
  - web citizen
  - git friendly

## 3.2 Starting the Jupyter Notebook

From a shell (Anaconda Prompt) or terminal we can start the notebook with:

```
jupyter notebook
```

This brings up the **dashboard** which will list any notebooks encountered in the current working directory.

You can either open an existing notebook or create a new one from the dashboard.

## 3.3 Jupyter Notebook Interface

- Menu
  - Keyboard Shortcuts: A list of keyboard shortcuts can be revealed by entering **h**.
- Toolbar
- Notebook area and cells

### 3.3.1 Classroom activities (do it together)

Click on “Help:User Interface Tour” menu item to start an interactive tour of the Notebook elements  
let’s do this!

### 3.3.2 Two keyboard input modes with Notebook

- *Edit Mode*: used to *edit* the content of a cell
  - Indicated by a green cell border and a prompt showing in the editor area
  - When a cell is in edit mode, the Cell Mode Indicator will change to reflect the cell’s state. This state is indicated by a small pencil icon on the top right of the interface. When the cell is in command mode, there is no icon in that location.
- *Command mode*: used to *manipulate* notebook cells
  - indicated by a grey cell border with a blue left margin

Switching between two modes: \* swiching from *Edit Mode* to *Command mode*: pressing **esc** \*  
swiching from *Command mode* to *Edit Mode*: pressing **enter**

### 3.3.3 Two Main Cell types

- Code Cell
- Markdown Cell

**Code cells** In command mode, **y** will set the cell to code. After that, pressing **Enter** will let you edit the code.

```
[1]: x = list(range(10)) # Shift-Enter to execute and move to the cell below
x
```

```
[1]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
[2]: x = list(range(10)) # Alt-Enter (option-Enter) to execute and inserts a new
↪ cell below.
x
```

```
[2]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
[3]: x.append(2)
x
```

```
[3]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 2]
```

```
[4]: x = range(10)
y = [ xi*3 for xi in x] # Ctrl-Return to execute but stay in the current cell
y
```

```
[4]: [0, 3, 6, 9, 12, 15, 18, 21, 24, 27]
```

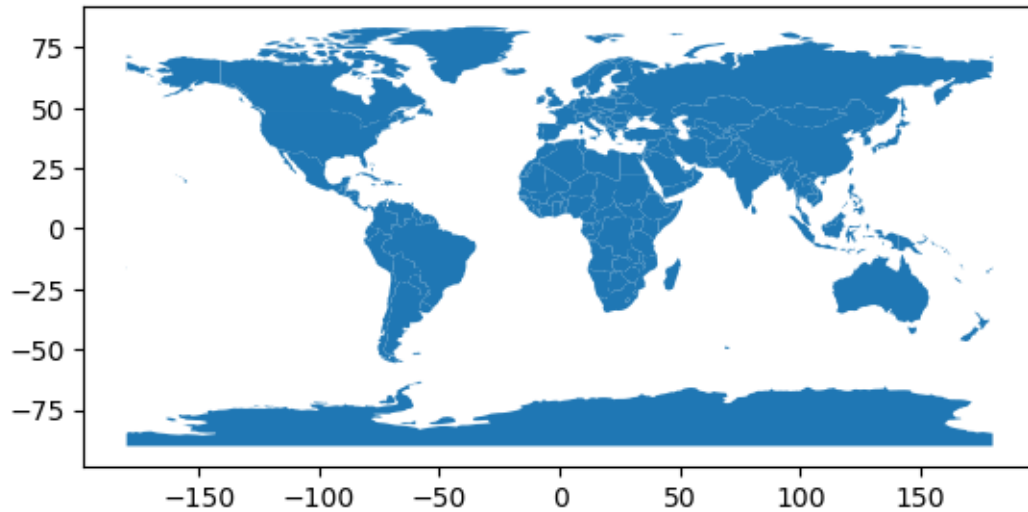
### 3.3.4 Comments on python code:

- start with #
- used to explain Python code
- used to make the code more readable.
- used to prevent execution when testing code.

```
[5]: #x = list(range(10)) # Shift-Enter to execute and move to the cell below
#x
```

```
[6]: # You will need to install geopandas first
import geopandas
world = geopandas.read_file(geopandas.datasets.get_path('naturalearth_lowres'))
world.plot();
```

```
/Users/wk0110/opt/anaconda3/lib/python3.9/site-
packages/geopandas/_compat.py:112: UserWarning: The Shapely GEOS version
(3.10.2-CAPI-1.16.0) is incompatible with the GEOS version PyGEOS was compiled
with (3.10.4-CAPI-1.16.2). Conversions between both will be slow.
warnings.warn(
```



### Order of the Code cells

- To signal that the Code cell was run, the Notebook will add a number [1]: to the left of the cell.
- This number increases with each run and shows if and in which order cells were run.
- Global State: Each Notebook has a single state that is shared between all cells, called the kernel.
  - Whenever you execute a cell, it modifies that state by running functions and setting variable values.

```
[7]: a = 1
```

```
[8]: a
```

```
[8]: 1
```

```
[9]: a = a + 1
```

```
[10]: a
```

```
[10]: 2
```

### 3.3.5 Markdown Cells and Markdown syntax

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

In command mode, `m` gives us a [Markdown](#) cell. After writing the text in Markdown syntax, press `shift+enter` to render the cell.

**list** an unordered list:

- first
- second
- third
  - nested one
  - nested two
- fourth

an ordered list:

1. first
2. third
  1. nested one
  2. nested two
3. fourth
4. fifth

## Headings

# 4 Heading 1

## 4.1 Heading 2

### 4.1.1 Heading 3

**Embedded code** You can embed code meant for illustration instead of execution in Python:  
(Reversed triple prime)

```
def f(x):  
    """a docstring"""  
    return x**2
```

```
pip install matplotlib
```

## Table

	Name
1	Tom
2	Jim

## Latex Equations with LaTeX

This is an in-line equation  $\hat{\beta} = (X'X)^{-1}X'y$  for the ordinary least squares estimator.

A display equation is done with

$$y = \rho W y + X\beta + \epsilon$$

**NOTE:** LaTeX will only render if [mathjax](#) is available via a network connection or if it has been installed locally.

#### 4.1.2 Command mode shortcuts :

- Basic navigation: **enter**, **shift-enter**, **up/k**, **down/j**
- Change Cell types: **y**, **m**, **1-6**,
- Cell creation: **a**, **b**
- Cell editing: **x**, **c**, **v**, **d**, **z**

#### 4.1.3 Kernel

Jupyter Notebook allows code to be run in a range of different programming languages. For each notebook document that a user opens, the web application starts a kernel that runs the code for that notebook. Each kernel is capable of running code in a single programming language and there are kernels available in the following languages:

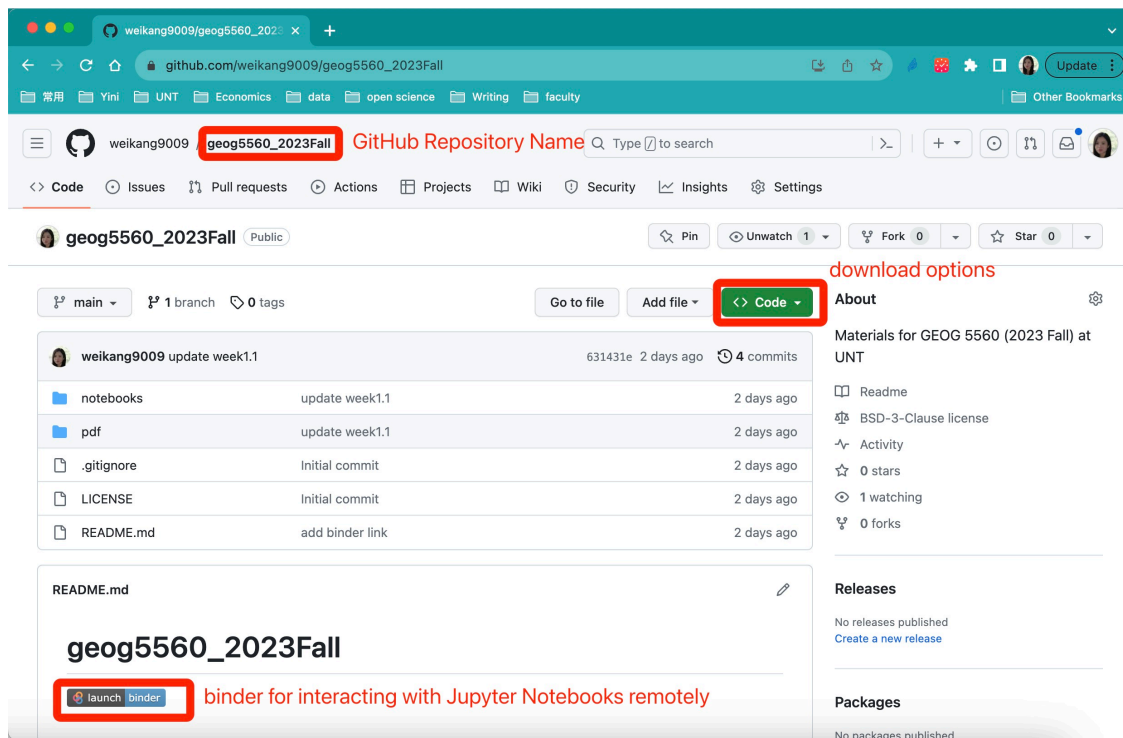
- Python(<https://github.com/ipython/ipython>)
- Julia (<https://github.com/JuliaLang/IJulia.jl>)
- R (<https://github.com/IRkernel/IRkernel>)
- Ruby (<https://github.com/minrk/iruby>)
- Haskell (<https://github.com/gibiansky/IHaskell>)
- Scala (<https://github.com/Bridgewater/scala-notebook>)
- node.js (<https://gist.github.com/Carreau/4279371>)
- Go (<https://github.com/takluyver/igo>)

## 5 GitHub

GitHub is a version-control platform for hosting development projects.

GitHub Repository (repo) for our course [https://github.com/weikang9009/geog5560\\_2023Fall](https://github.com/weikang9009/geog5560_2023Fall)





We will use GitHub and Git to access our course materials, which will be updated every few days.

## 5.1 Additional readings

- [Anaconda User Guide](#)
- [Jupyter Notebook Users Manual](#)
- [Markdown](#)

## 6 Next class

- Topics: Program, Variables, Operators
- before the class, read Chapters 1, 2 of “Downey, A.B. (2015) Think Python: How to Think Like a Computer Scientist. 2nd Edition. Green Tea Press, Needham, Massachusetts.”