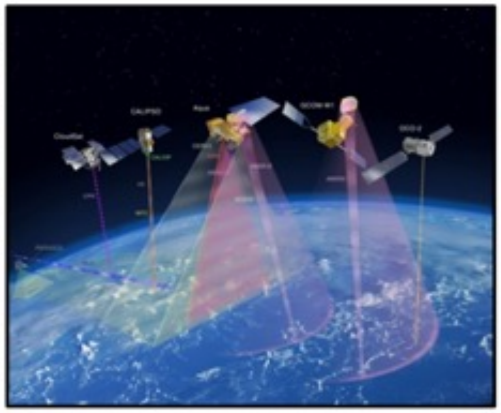
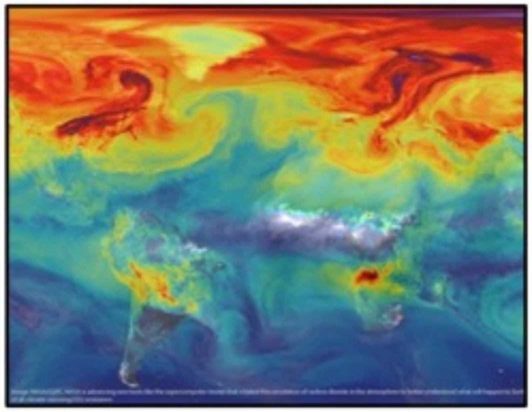



Welcome to my Class!

Geo Data Science with Python
(GEOS-5984/4984)

Before class, I will to be here a bit earlier to answer your questions and play one or two songs before class. Since we can't see each other's faces this semester, we can at least enjoy some music. Please suggest me your favorite songs. Non-English songs are very welcome!

Computer Login:	Username:	your VT-PID
	Password:	<all your initials> + "hokie22"

Personal Introduction

Name, field of study, your favorite type of music

Survey

Before end of today’s classes, write on paper:

- **Colored paper, with name:**
  What is your previous coding experience (none/medium/strong, which programming languages?)

- **White paper, anonymous**:
  What are your expectations for the course?

Today

- Survey: please fill out before end of class
- Intro & Syllabus
- Jupyter Notebooks

Intro

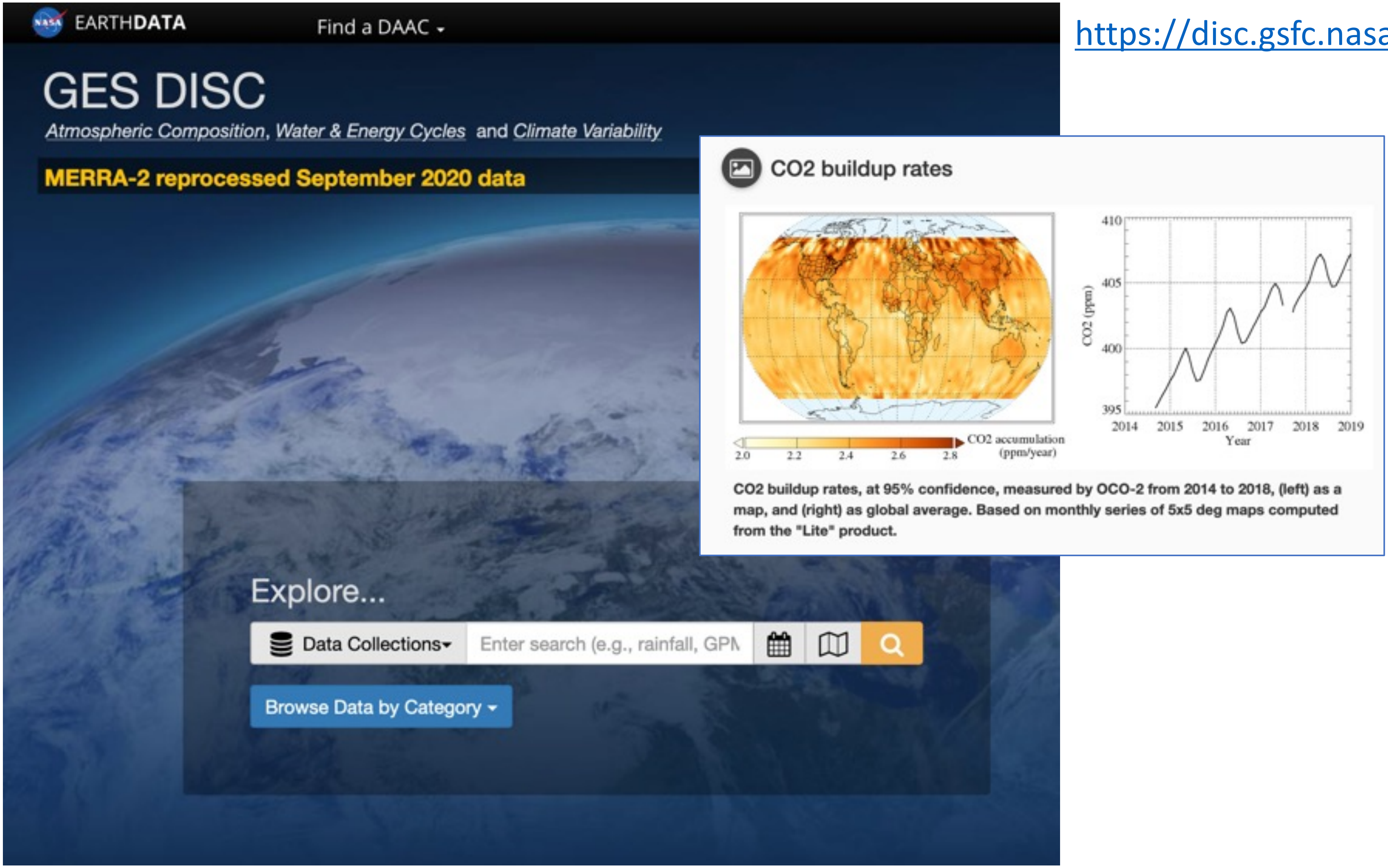
- Syllabus / Course Information
  - Canvas > Syllabus
  - https://canvas.vt.edu/courses/157579/assignments/syllabus
  - Learning Objectives, Outcomes & Philosophy >>> next slides
  - Student hours
  - Course schedule
  - Material
  - Academic Integrity & Plagiarism

Learning Objectives: Python Basics

```
Python 3.1.4 (default, Jun 12 2011, 15:05:44) [MSC v.1500 32 bit (Intel)]
32
Type "copyright", "credits" or "license()" for more information.
>>> import sys
>>> k = 2+2
['where', 'do', 'you', 'want', 'to', 'go', 'today?']
>>> print(a[0])
where
```

Learning Objectives: Getting Data

https://disc.gsfc.nasa.gov/



EARTHDATA

Find a DAAC

GES DISC

Atmospheric Composition, *Water & Energy Cycles* and *Climate Variability*

MERRA-2 reprocessed September 2020 data

CO2 buildup rates

CO2 buildup rates, at 95% confidence, measured by OCO-2 from 2014 to 2018, (left) as a map, and (right) as global average. Based on monthly series of 5x5 deg maps computed from the "Lite" product.

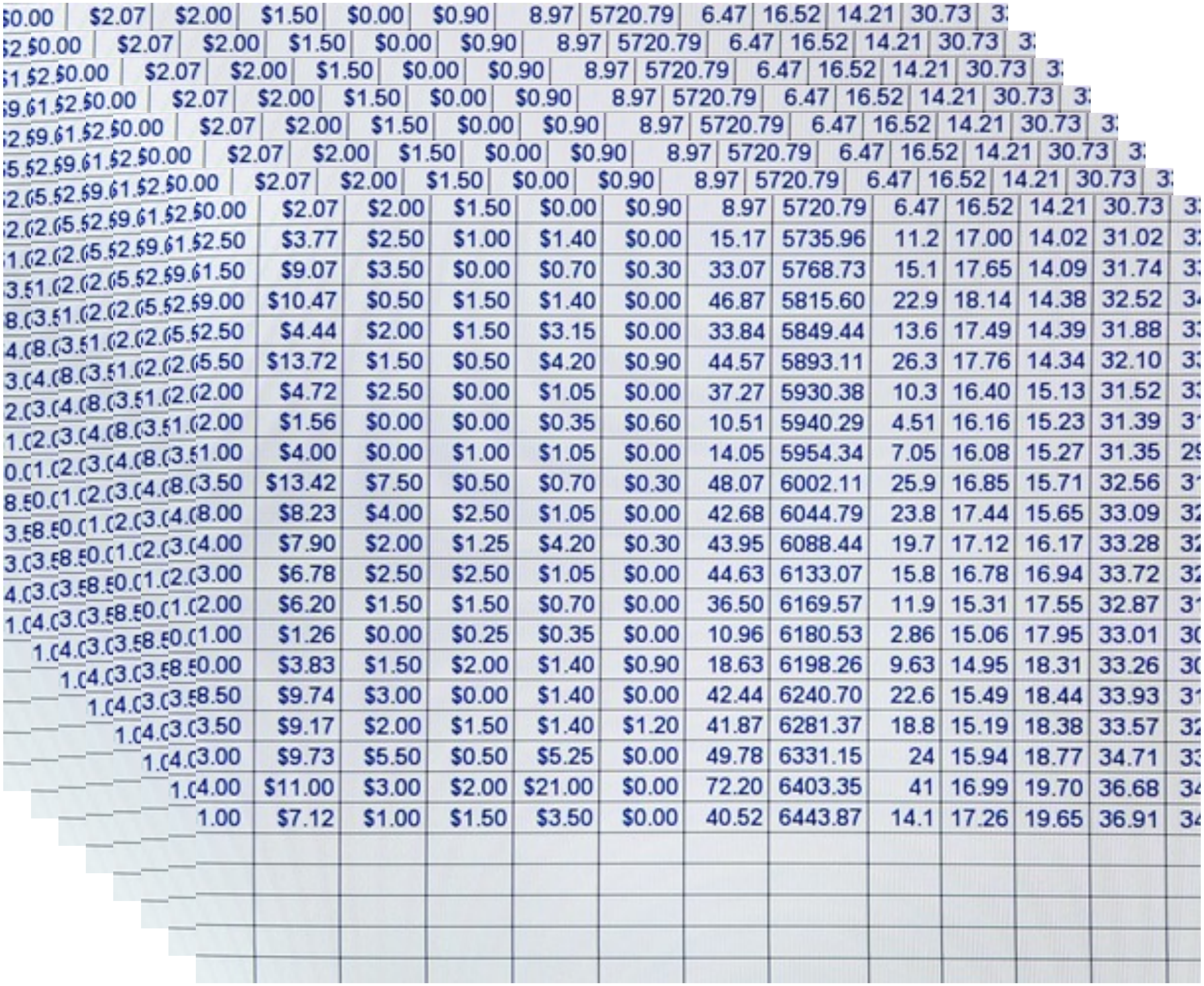
Explore...

Data Collections

Enter search (e.g., rainfall, GPM

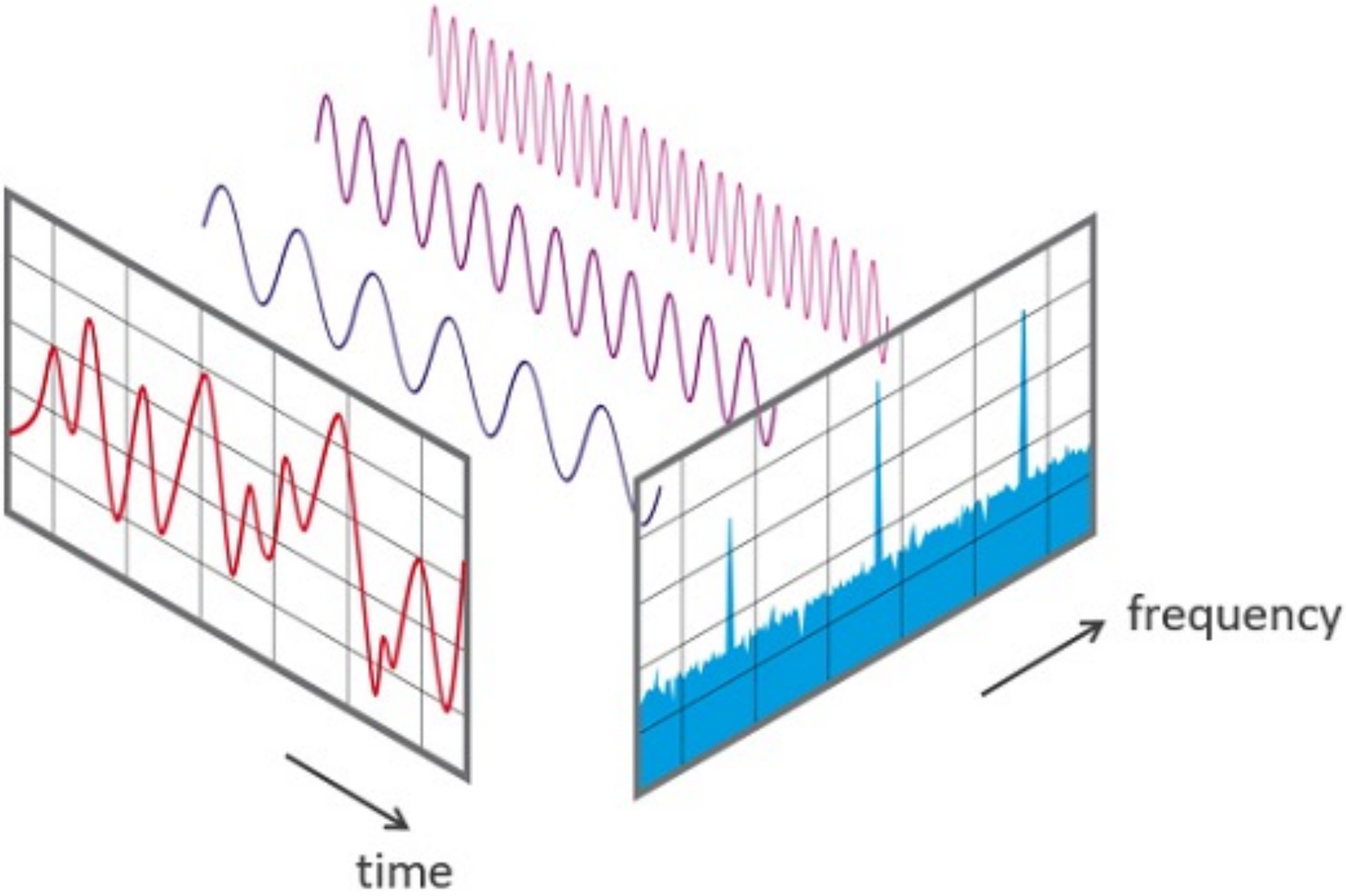
Browse Data by Category

Learning Objectives: Lot’s of Data

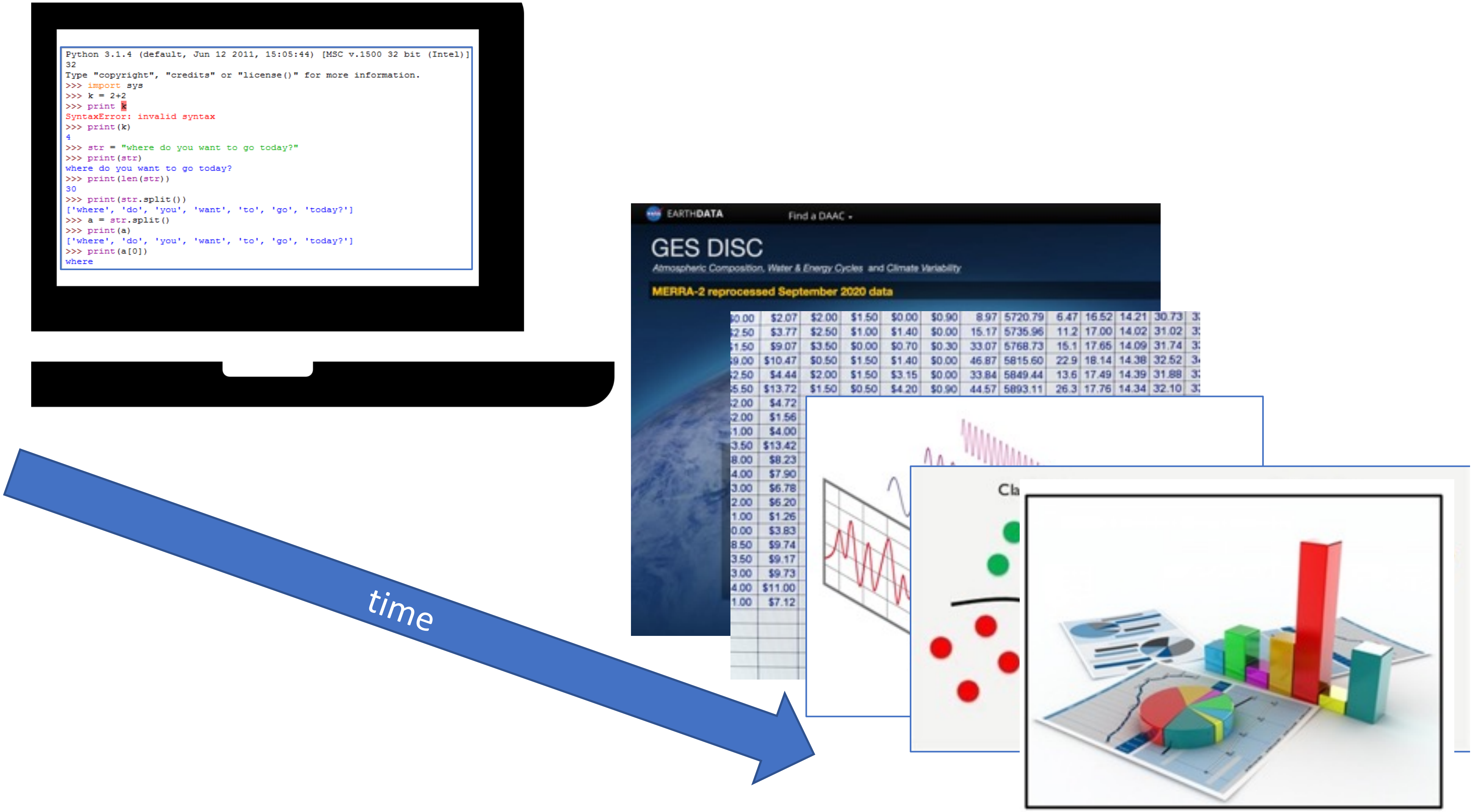


0.00	$2.07	$2.00	$1.50	$0.00	$0.90	8.97	5720.79	6.47	16.52	14.21	30.73	3
2.50	$3.77	$2.50	$1.00	$1.40	$0.00	15.17	5735.96	11.2	17.00	14.02	31.02	3
1.50	$9.07	$3.50	$0.00	$0.70	$0.30	33.07	5768.73	15.1	17.65	14.09	31.74	3
9.00	$10.47	$0.50	$1.50	$1.40	$0.00	46.87	5815.60	22.9	18.14	14.38	32.52	3
2.50	$4.44	$2.00	$1.50	$3.15	$0.00	33.84	5849.44	13.6	17.49	14.39	31.88	3
5.50	$13.72	$1.50	$0.50	$4.20	$0.90	44.57	5893.11	26.3	17.76	14.34	32.10	3
2.00	$4.72	$2.50	$0.00	$1.05	$0.00	37.27	5930.38	10.3	16.40	15.13	31.52	3
2.00	$1.56	$0.00	$0.00	$0.35	$0.60	10.51	5940.29	4.51	16.16	15.23	31.39	3
1.00	$4.00	$0.00	$1.00	$1.05	$0.00	14.05	5954.34	7.05	16.08	15.27	31.35	2
3.50	$13.42	$7.50	$0.50	$0.70	$0.30	48.07	6002.11	25.9	16.85	15.71	32.56	3
8.00	$8.23	$4.00	$2.50	$1.05	$0.00	42.68	6044.79	23.8	17.44	15.65	33.09	3
4.00	$7.90	$2.00	$1.25	$4.20	$0.30	43.95	6088.44	19.7	17.12	16.17	33.28	3
3.00	$6.78	$2.50	$2.50	$1.05	$0.00	44.63	6133.07	15.8	16.78	16.94	33.72	3
2.00	$6.20	$1.50	$1.50	$0.70	$0.00	36.50	6169.57	11.9	15.31	17.55	32.87	3
1.00	$1.26	$0.00	$0.25	$0.35	$0.00	10.96	6180.53	2.86	15.06	17.95	33.01	3
0.00	$3.83	$1.50	$2.00	$1.40	$0.90	18.63	6198.26	9.63	14.95	18.31	33.26	3
8.50	$9.74	$3.00	$0.00	$1.40	$0.00	42.44	6240.70	22.6	15.49	18.44	33.93	3
3.50	$9.17	$2.00	$1.50	$1.40	$1.20	41.87	6281.37	18.8	15.19	18.38	33.57	3
3.00	$9.73	$5.50	$0.50	$5.25	$0.00	49.78	6331.15	24	15.94	18.77	34.71	3
4.00	$11.00	$3.00	$2.00	$21.00	$0.00	72.20	6403.35	41	16.99	19.70	36.68	3
1.00	$7.12	$1.00	$1.50	$3.50	$0.00	40.52	6443.87	14.1	17.26	19.65	36.91	3

Learning Objectives: Data Science



Outcome: Final Project



```
Python 3.1.4 (default, Jun 12 2011, 15:05:44) [MSC v.1500 32 bit (Intel)]
32
Type "copyright", "credits" or "license()" for more information.
>>> import sys
>>> k = 2+2
>>> print k
SyntaxError: invalid syntax
>>> print(k)
4
>>> str = "where do you want to go today?"
>>> print(str)
where do you want to go today?
>>> print(len(str))
30
>>> print(str.split())
['where', 'do', 'you', 'want', 'to', 'go', 'today?']
>>> a = str.split()
>>> print(a)
['where', 'do', 'you', 'want', 'to', 'go', 'today?']
>>> print(a[0])
where
```

EARTHDATA
Find a DAAC
GES DISC
Atmospheric Composition, Water & Energy Cycles and Climate Variability
MERRA-2 reprocessed September 2020 data
Cla
time

Outcome: Final Project

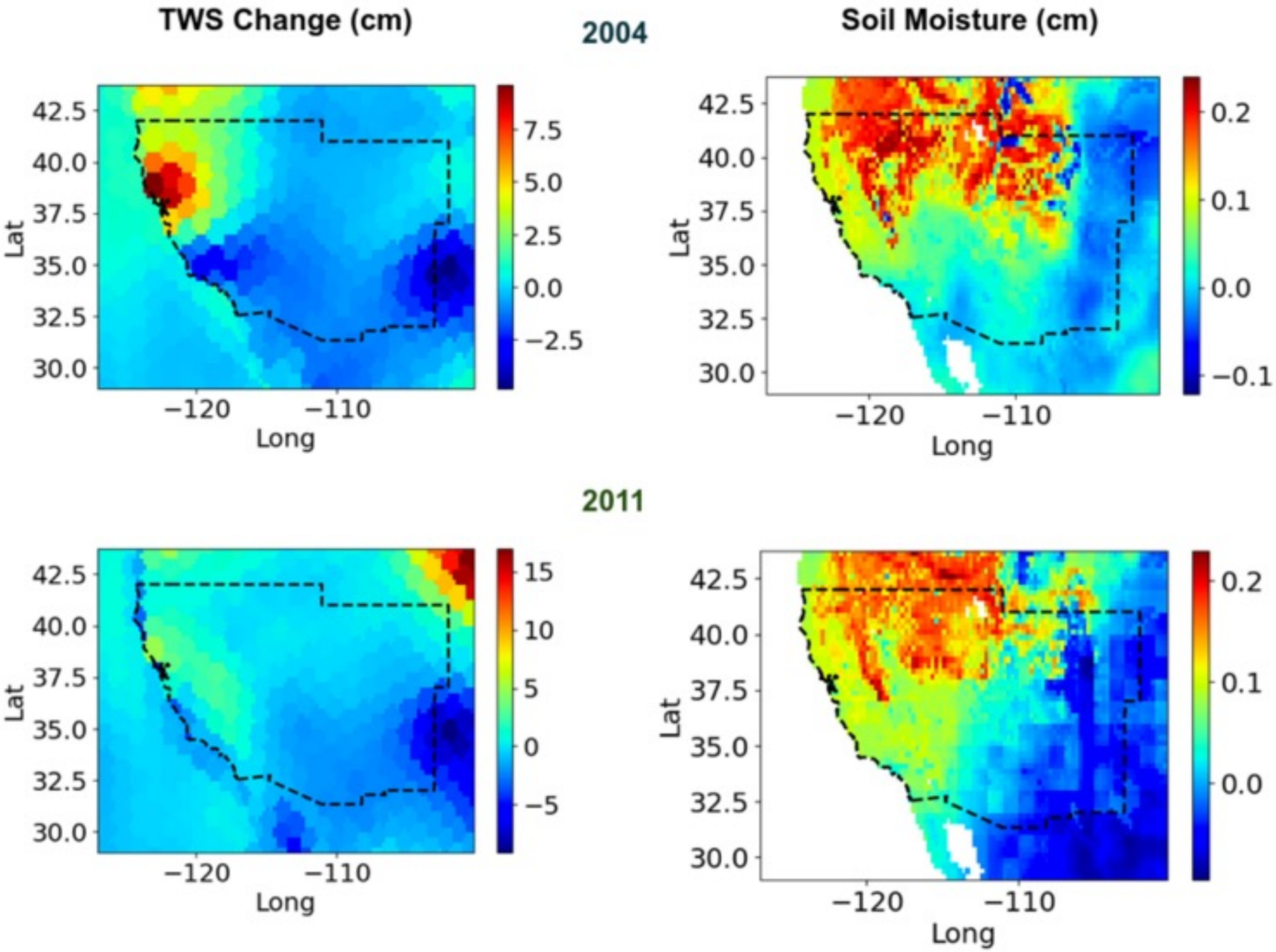


Figure 1. Total water storage change and Soil Moisture change in the first month of study (Jan., 2004) and the last month of study (Dec., 2011).

Curtesy: Sonia Zehsaz and Mohammad Khorrami

Outcome: Final Project

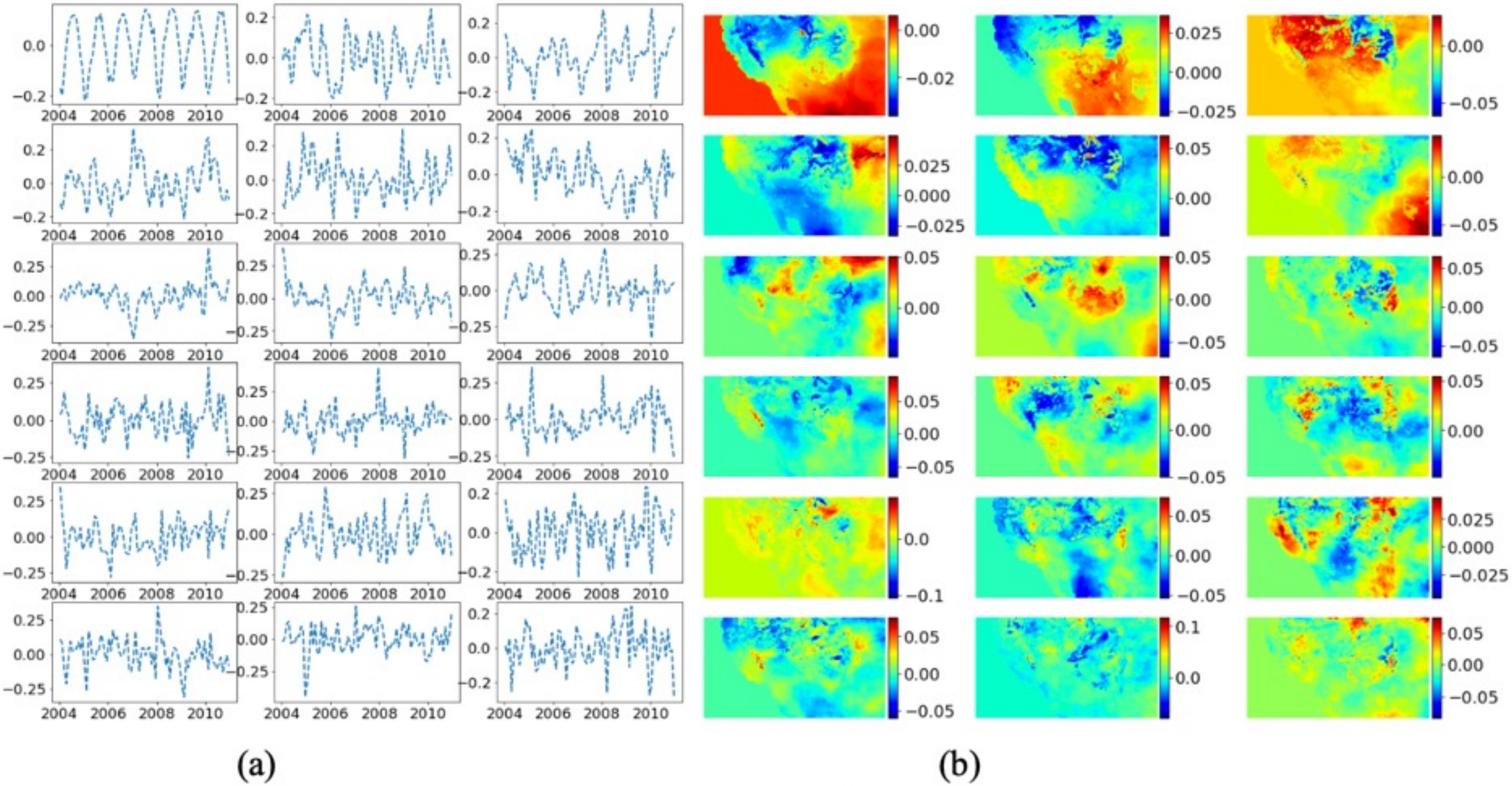
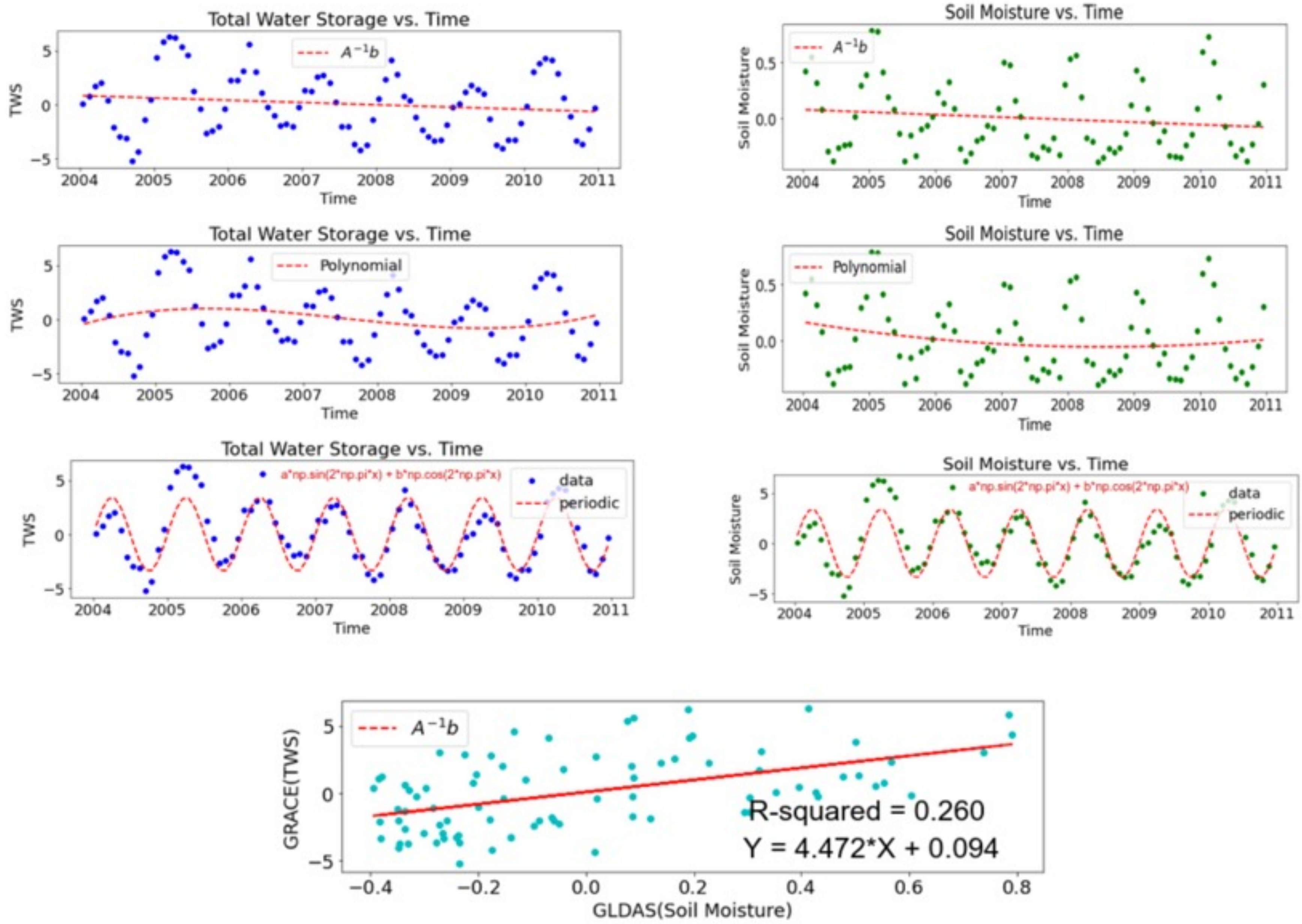


Figure 4. the main 18 features of soil moisture data (a) and their times series (b).

Curtesy: Sonia Zehsaz and Mohammad Khorrami

Outcome: Final Project



Curtesy: Sonia Zehsaz and Mohammad Khorrami

Course Sections

(about 4 weeks each)

1. Python Basics
2. Working with big (geo) data
3. Machine Learning basics for Geosciences
4. Final Project

> See details in Syllabus

Course Philosophy

- Focus on your learning style: listening, reading, watching.
- Coding is for everyone and useful for everything!
- Practice. Practice. Practice.
- Don’t try to avoid the mistake. Make it (when you practice) and learn from it!
- Break down problems to simpler ones.
- Learn together and from each other.
- Ask me, and ask again!

Intro

- Syllabus / Course Information
  - Canvas > Syllabus
  - https://canvas.vt.edu/courses/157579/assignments/syllabus
  - Learning Objectives, Outcomes & Philosophy
  - Course schedule: *preliminary*
  - Student hours: *Tuesdays after class*
  - Material: *next week*
  - Academic Integrity & Plagiarism: *please read carefully!*

NEW Computer Login

Username: <yourPID>

Password: <all your initials> + “hokie22”

After login, change your password:

Ctrl + Alt + Del

Click on Change Password

Make sure to backup local data and logout after class!

Writing & Executing Python Code

A. **Interactive mode**:
type commands one-at-a-time in the interactive shell

B. **Script Mode**:
execute the commands from a script file (sequentially), composition in an IDE

C. **Notebooks:**
browser-based interface IDE **and** interactive document

VIRGINIA TECH™

COLLEGE OF SCIENCE
VIRGINIA TECH™

Introducing
Jupyter Notebooks

jupyter

Prof. Susanna Werth

Learning Objectives

1. Name and install software to use Jupyter
2. Explain what Jupyter Notebook and Lab is
3. Name different components of a Jupyter Notebook
4. Write and execute basic Python code within a Jupyter Notebook

The Jupyter Project

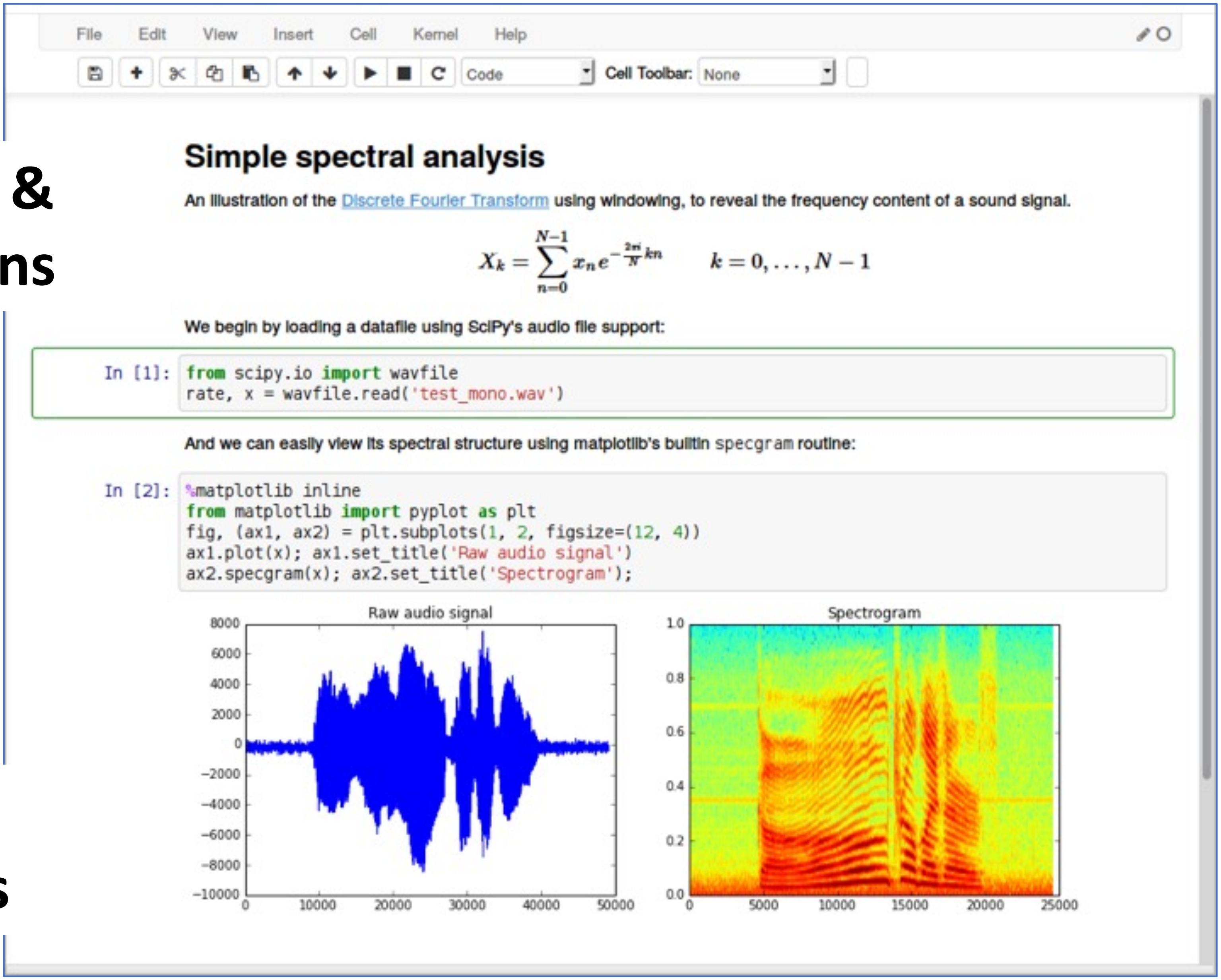
Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

The Jupyter Project

- Open source effort that evolved from the IPython project (started 2014) to support interactive data science and computing.
- Now supports over 40 other **interpreted** programming languages, including: Python, R, Scalar, Julia, Ruby, Java, Matlab, Octave, Processing, Scheme, etc.

  - Main page: jupyter.org
  - Docs: jupyter.org/documentation
  - Kernels: github.com/jupyter/jupyter/wiki/Jupyter-kernels

The Jupyter Notebook



Simple spectral analysis

An illustration of the Discrete Fourier Transform using windowing, to reveal the frequency content of a sound signal.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}kn} \qquad k = 0,\dots,N-1$$

We begin by loading a datafile using SciPy's audio file support:

```
In [1]: from scipy.io import wavfile
        rate, x = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's builtin `specgram` routine:

```
In [2]: %matplotlib inline
        from matplotlib import pyplot as plt
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
        ax1.plot(x); ax1.set_title('Raw audio signal')
        ax2.specgram(x); ax2.set_title('Spectrogram');
```

Equations & descriptions

Code

Output, Graphics

Jupyter Notebooks

App

- Jupyter Notebook is a browser-based interface & Integrated Development Environment (IDE)
- Allows to write code, navigate files on our computer, inspect variables and more.
- Allows to write code in many programming languages, including Python

and **Interactive document**

- Combines live code blocks and output, equations, visualizations and narrative text.
- Formatted text blocks can describe what the code does
- Integrates various media.
- Serve as **interactive textbooks** due to the executable and editable code/text blocks.

Jupyter Notebooks

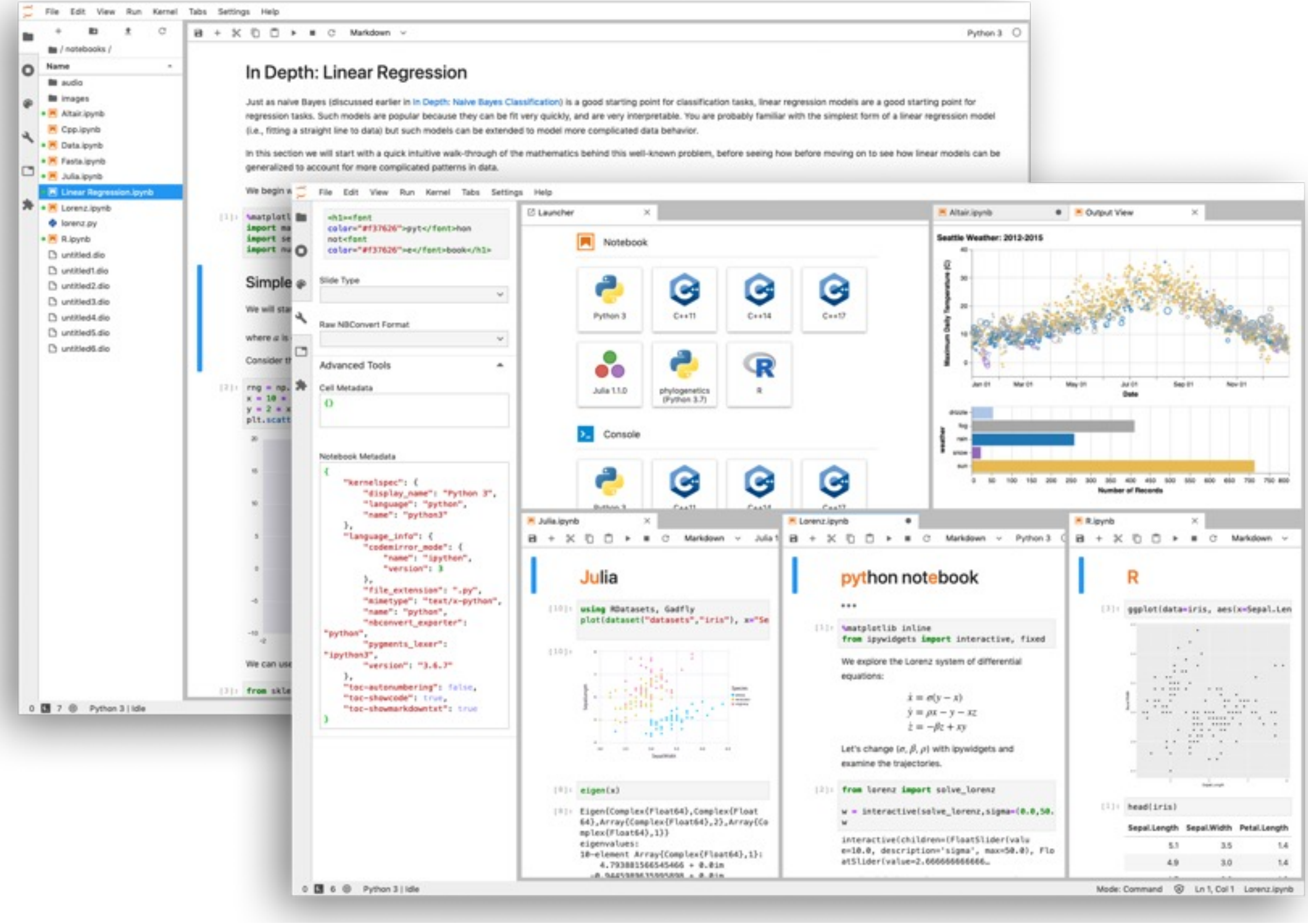
Advantages:

- More accessible than scripted Python programs.
- Easy onramp for coding-unexperienced people: to play with and understand your code or your project.
- Promote hands-on learning and experimentation.
- Convenient because they run in a browser.
- Has become a popular user interface for cloud computing.

Organize and Document Workflows Using Jupyter Notebook Files

- Connecting your entire **workflow** including accessing the data, processing methods and outputs
- **Documentation** of data inputs, code for analysis and visualization, and results – all within one file that can be easily **shared** with others.
- An important part of **open reproducible science**.
- You can also export notebook files to HTML and PDF formats for easy sharing

Jupyter Lab



In Depth: Linear Regression

Just as naive Bayes (discussed earlier in In Depth: Naive Bayes Classification) is a good starting point for classification tasks, linear regression models are a good starting point for regression tasks. Such models are popular because they can be fit very quickly, and are very interpretable. You are probably familiar with the simplest form of a linear regression model (i.e., fitting a straight line to data) but such models can be extended to model more complicated data behavior.

In this section we will start with a quick intuitive walk-through of the mathematics behind this well-known problem, before seeing how before moving on to see how linear models can be generalized to account for more complicated patterns in data.

```
<h1><font
color="#f37626">pyt</font>hon
not<font
color="#f37626">e</font>book</h1>
```

Slide Type

Raw NBConvert Format

Advanced Tools

Cell Metadata

```
{}
```

Notebook Metadata

```
{
    "kernelspec": {
        "display_name": "Python 3",
        "language": "python",
        "name": "python3"
    },
    "language_info": {
        "codemirror_mode": {
            "name": "ipython",
            "version": 3
        },
        "file_extension": ".py",
        "mimetype": "text/x-python",
        "name": "python",
        "nbconvert_exporter":
"python",
        "pygments_lexer":
"ipython3",
        "version": "3.6.7"
    },
    "toc-autonumbering": false,
    "toc-showcode": true,
    "toc-showmarkdowntxt": true
}
```

Julia

```
[10]: using RDatasets, Gadfly
      plot(dataset("datasets","iris"), x="Se
```

```
[8]: eigen(a)
```

```
[8]: Eigen{Complex{Float64},Complex{Float
     64},Array{Complex{Float64},2},Array{Co
     mplex{Float64},1}}
     eigenvalues:
     10-element Array{Complex{Float64},1}:
         4.793881566545466 + 0.0im
```

python notebook

***

```
[1]: %matplotlib inline
     from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
[2]: from lorenz import solve_lorenz

     w = interactive(solve_lorenz,sigma=(0.0,50.
     w
```

```
interactive(children=(FloatSlider(valu
e=10.0, description='sigma', max=50.0), Flo
atSlider(value=2.666666666666…
```

R

```
[3]: ggplot(data=iris, aes(x=Sepal.Len
```

```
[1]: head(iris)
```

Sepal.Length	Sepal.Width	Petal.Length
5.1	3.5	1.4
4.9	3.0	1.4

Jupyter Lab

- **Next-generation** app from the Jupyter project
- Provides more flexible environment for working with Jupyter notebooks
- Working with several notebooks at once.
- Each notebook can have its own associated environment (e.g. conda environment or a particular programming language).
- Views allow for several separate panels, to see more of what you are working on in one view.

Resources for the Introduction

- Scott D. Peckham, UC Colorado, BALTO Jupyter Notebook workshop slides
- Jupyter project pages: jupyter.org
- Jupyter Notebook documenation pages: https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/index.html
- Jupyter (IPython) notebooks features: https://arogozhnikov.github.io/2016/09/10/jupyter-features.html

Python

- In this course, we use Jupyter Notebook with Python.
- Python is a widely used programming language in the sciences and provides strong functionality for working with a variety of data types and formats.

www.python.org

Software:
The Anaconda Python Distribution

- The Anaconda Python distribution for individuals is **open-source**, **free,** and very **widely used**.
- Anaconda supports: MacOS, Linux and Windows.
- Other Python distributions exist, and your computer (especially Linux or Mac) may come with a basic Python distribution.
- Anaconda is a solid choice and really does make it easier to use Python on your computer, especially when you are working with many packages that could potentially conflict with one another.
  - **Main page**: https://www.anaconda.org
  - **Download**: https://www.anaconda.com/products/distribution

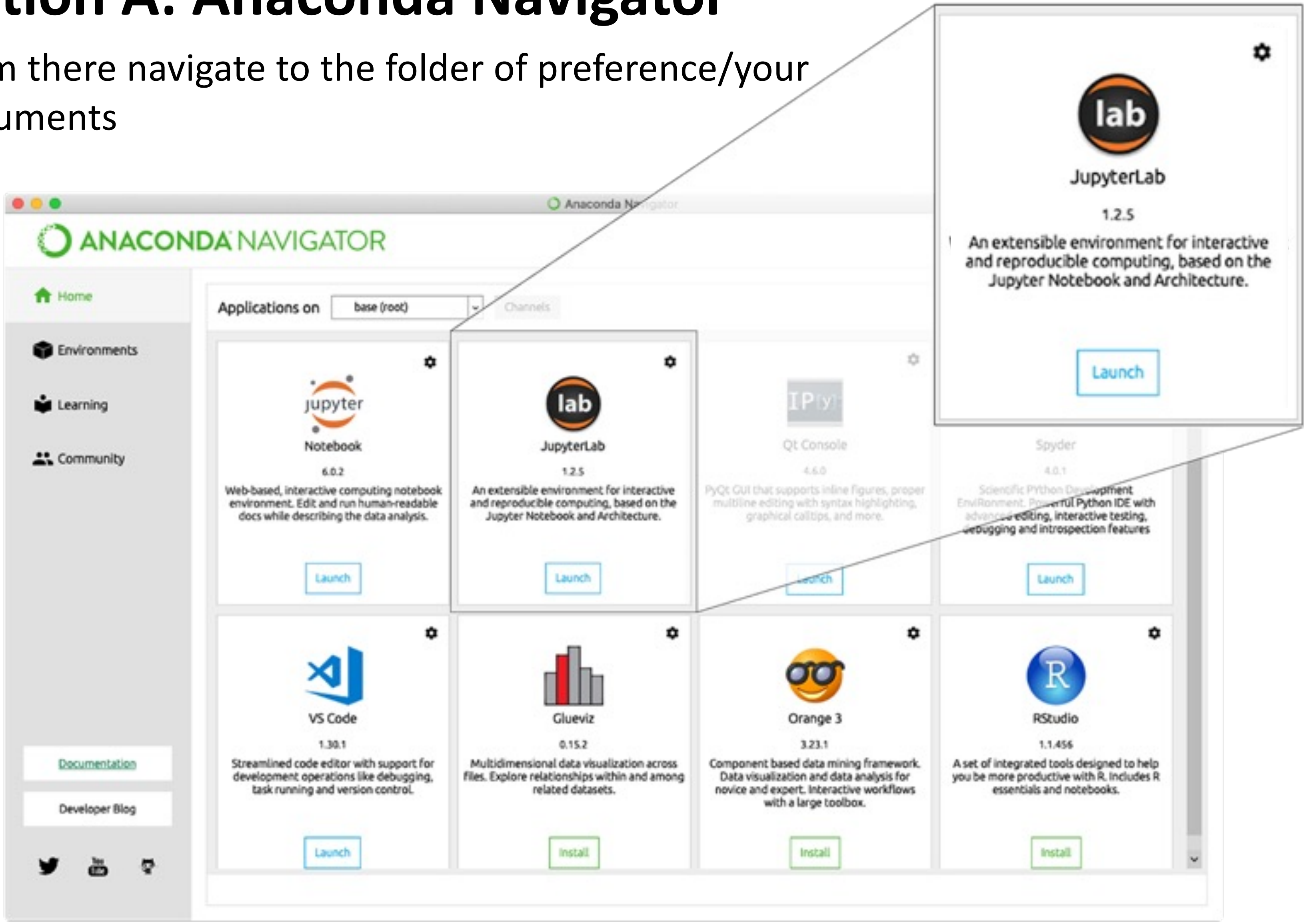
Jupyter Notebook Basics

Prof. Susanna Werth

Launch Jupyter Lab

Option A: Anaconda Navigator

From there navigate to the folder of preference/your documents



Notebook: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

JupyterLab: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Qt Console: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Spyder: Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

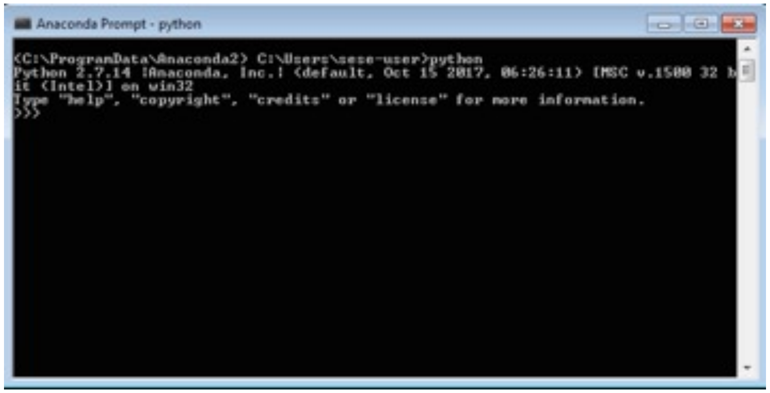
VS Code: Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz: Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3: Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

RStudio: A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Launch Jupyter Lab



```
(C:\ProgramData\Anaconda2) C:\Users\sese-user>python
Python 2.7.14 |Anaconda, Inc.| (default, Oct 15 2017, 06:26:11) [MSC v.1500 32 b
it (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Option B: Command line (Terminal)

1. Navigate to the data directory of your choice

    Unix Terminal/Console:

    ```
    cd ~/Documents
    ```

    Windows Command Prompt:

    ```
    cd /D %userprofile%\Documents
    ```

2. Start Jupyter server by typing:

    ```
    jupyter lab
    ```

Command prompts

Useful file system-related commands

Function	shell command (Linux, Mac)	respective command (Windows)
display current directory	**pwd**	cd
display content of current directory	**ls**	dir
go to ‘directory’	**cd** ‘directory’	cd ‘directory’
create directory	**mkdir** ‘directory’	md ‘directory’
copy file	**cp** ‘file’	copy ‘file’
delete file	**rm** ‘file’	del ‘file’
display file	**cat** ‘file’	type ‘file’

Launch Jupyter Lab

- We continue, once everybody has their Jupyter lab launched
- Navigate to folder “Documents”

Note on Classroom Computers

- To start Jupyter lab (or spyder), you do not need open anaconda, but can type ‘jupyter lab’ or ‘spyder’ into the command line.
- Remember to browse in private mode, if you login to accounts!
- Remember, this is only local. You can download files from the internet and work with them. **Don’t forget to take a copy of your work with you**, or upload it to your Google drive or a private GitHub repository

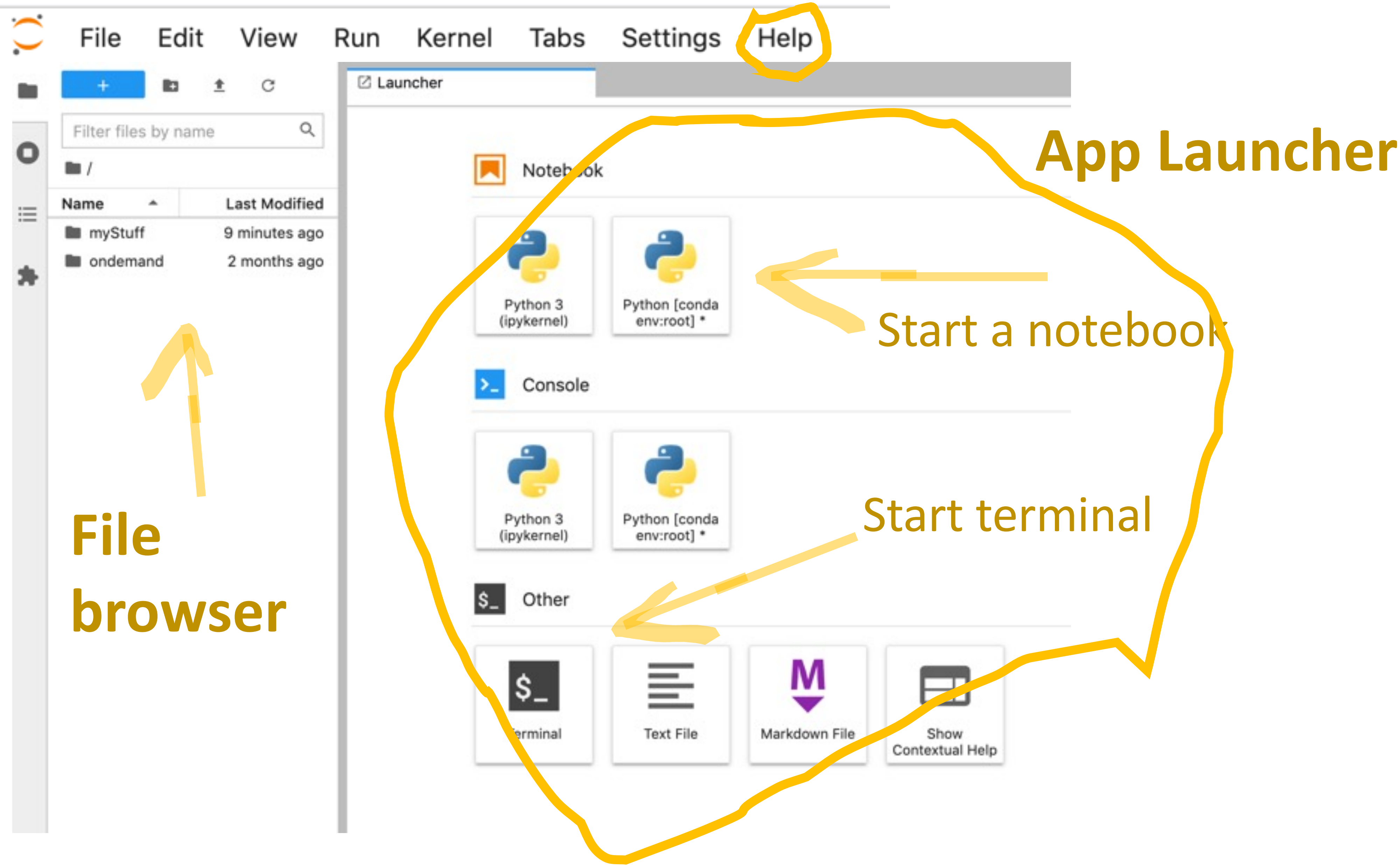
Tutorial: Jupyter Lab

Objectives

- Launch jupyter lab (command line or anaconda)
- Navigate through the app
- Create & use notebook
- Work with cells & basic shortcuts
- Correctly shutting down a server

... follow my steps in Jupyter Lab ...

Jupyter Lab App Navigation



Name	Last Modified
myStuff	9 minutes ago
ondemand	2 months ago

App Launcher

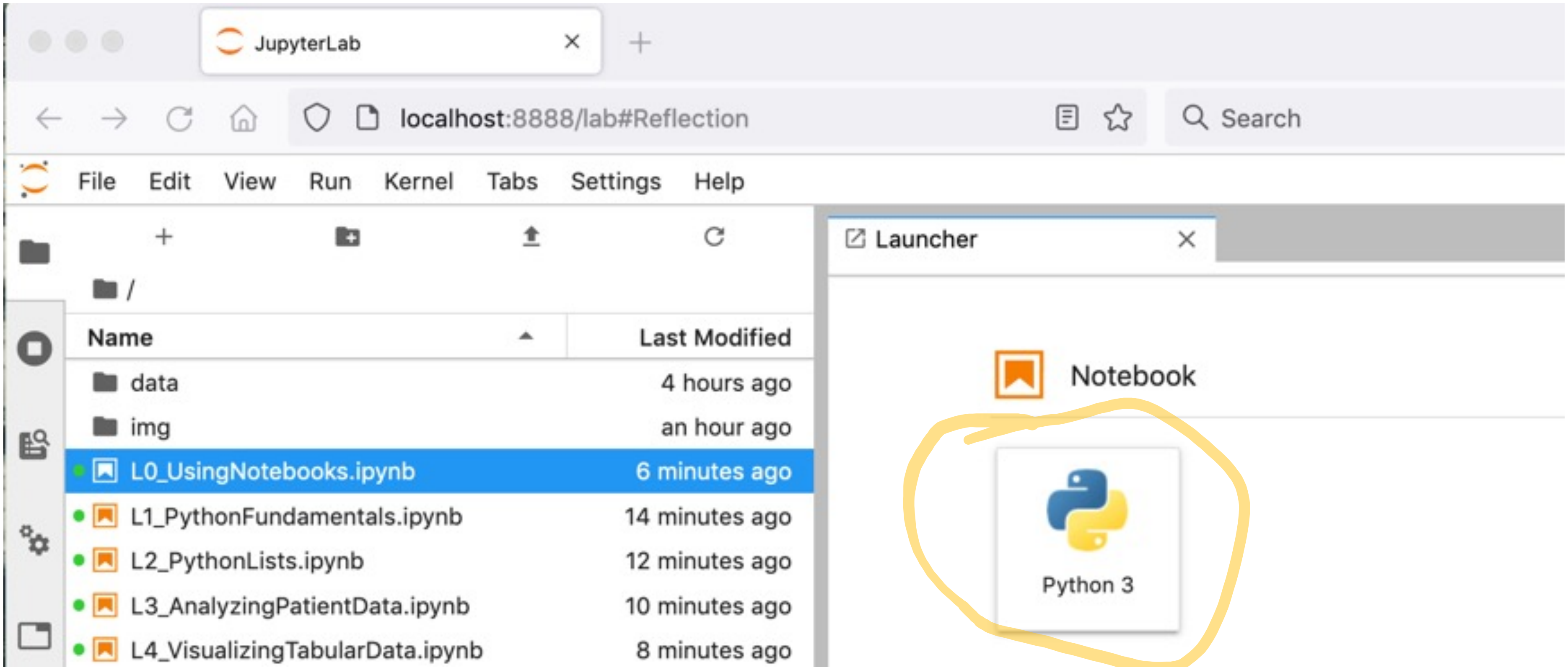
Start a notebook

Start terminal

File browser

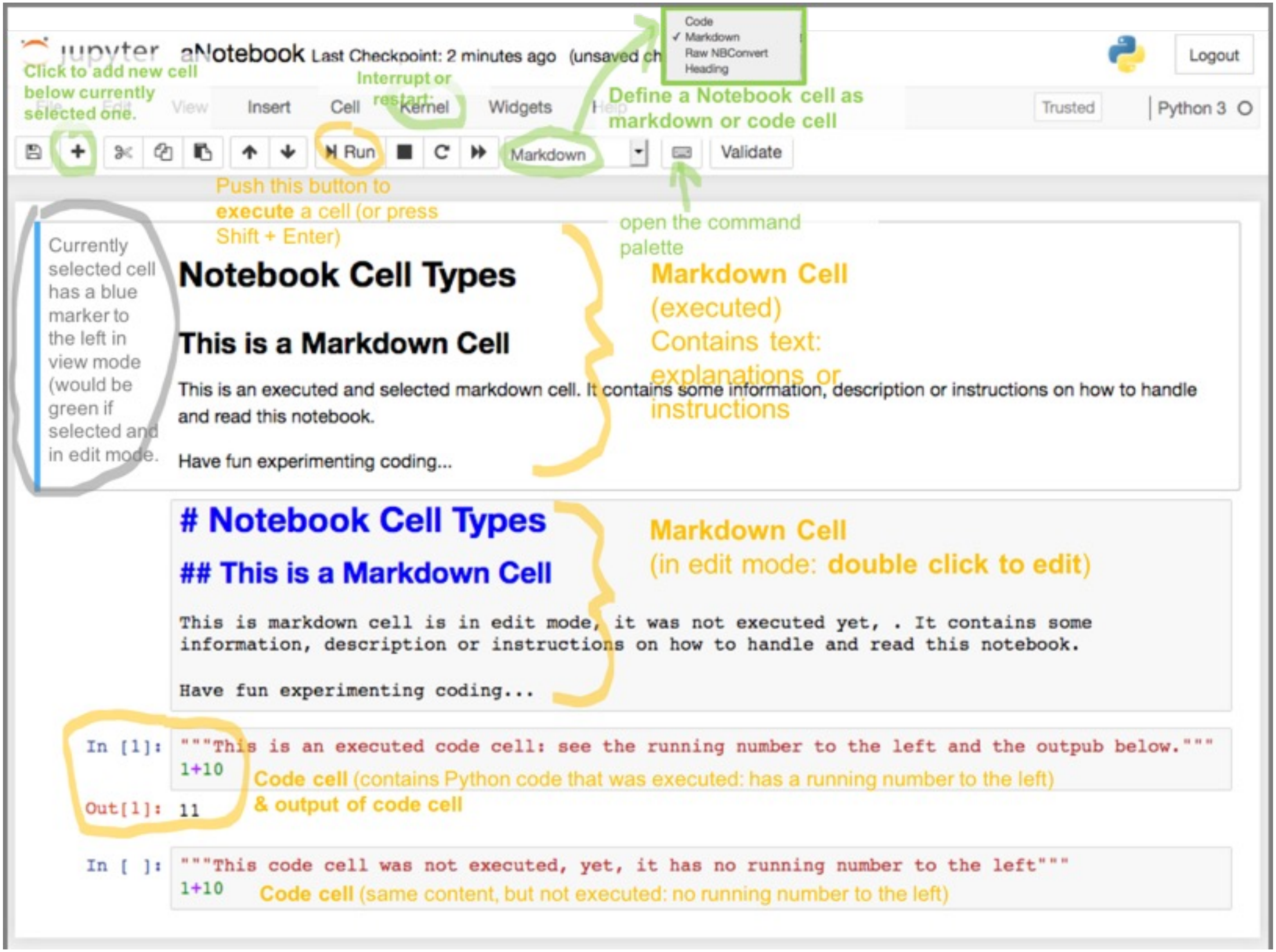
Launch Jupyter Notebooks

Now launch the notebook by clicking on the Notebook “Launcher” button:



Name	Last Modified
data	4 hours ago
img	an hour ago
L0_UsingNotebooks.ipynb	6 minutes ago
L1_PythonFundamentals.ipynb	14 minutes ago
L2_PythonLists.ipynb	12 minutes ago
L3_AnalyzingPatientData.ipynb	10 minutes ago
L4_VisualizingTabularData.ipynb	8 minutes ago

Jupyter Notebooks



Click to add new cell below currently selected one.

Interrupt or restart:

Define a Notebook cell as markdown or code cell

Push this button to **execute** a cell (or press Shift + Enter)

open the command palette

Currently selected cell has a blue marker to the left in view mode (would be green if selected and in edit mode.

Notebook Cell Types

This is a Markdown Cell

This is an executed and selected markdown cell. It contains some information, description or instructions on how to handle and read this notebook.

Have fun experimenting coding...

Markdown Cell (executed) Contains text: explanations or instructions

```
# Notebook Cell Types
## This is a Markdown Cell

This is markdown cell is in edit mode, it was not executed yet, . It contains some
information, description or instructions on how to handle and read this notebook.

Have fun experimenting coding...
```

Markdown Cell (in edit mode: **double click to edit**)

```
In [1]: """This is an executed code cell: see the running number to the left and the outpub below."""
        1+10

Out[1]: 11
```

Code cell (contains Python code that was executed: has a running number to the left)

& output of code cell

```
In [ ]: """This code cell was not executed, yet, it has no running number to the left"""
        1+10
```

Code cell (same content, but not executed: no running number to the left)

Important Keyboard Shortcuts

Esc will take you into command mode where you can navigate around your notebook with arrow keys.

While in command mode:

A to insert a new cell above the current cell, B to insert a new cell below.

M to change the current cell to Markdown, Y to change it back to code

D + D (press the key twice) to delete the current cell

Enter will take you from command mode back into edit mode for the given cell.

Shift + Tab will show you the Docstring (documentation) for the the object you have just typed in a code cell – you can keep pressing this short cut to cycle through a few modes of documentation.

Ctrl + Shift + - will split the current cell into two from where your cursor is.

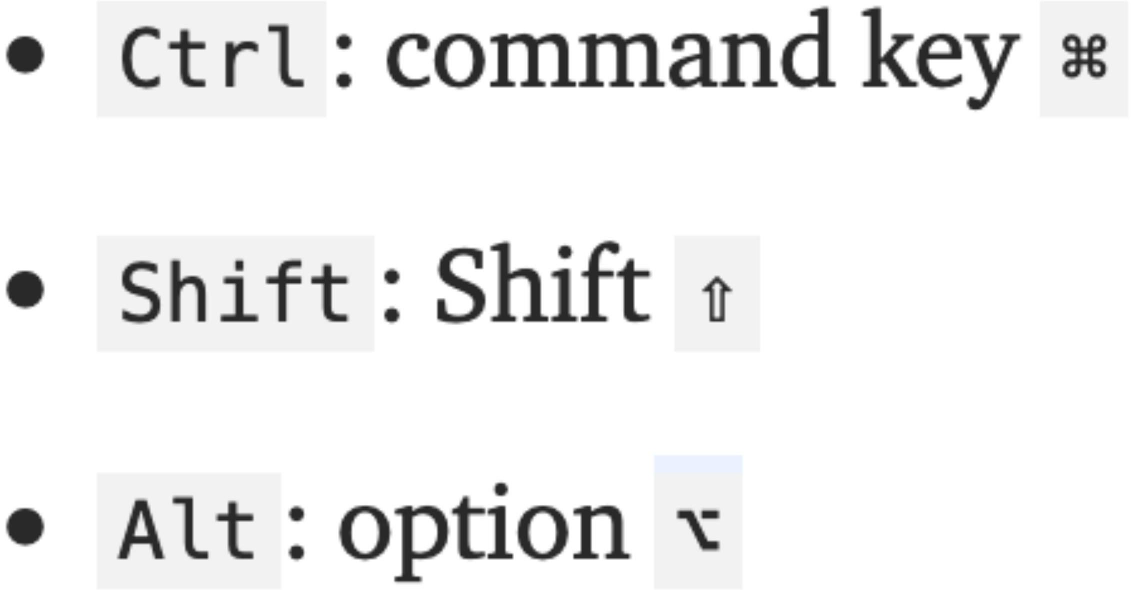
Esc + F Find and replace on your code but not the outputs.

Esc + O Toggle cell output.

More at: https://cheatography.com/weidadeyue/cheat-sheets/jupyter-notebook/

Keyboard Shortcuts

- Shortcuts are for **Windows** and **Linux** users
- For **Mac** users they’re different buttons for **Ctrl, Shift and Alt:**



- `Ctrl`: command key `⌘`
- `Shift`: Shift `⇧`
- `Alt`: option `⌥`

Resources for Notebook Intro

- Jupyter Notebook documenation pages: https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/index.html
- Jupyter Notebook Tutorial: Introduction, Setup, and Walkthrough from Corey Schafer: https://youtu.be/HW29067qVWk
- Jupyter (IPython) notebooks features: https://arogozhnikov.github.io/2016/09/10/jupyter-features.html
- Interactive IPython Notebook Overview http://quasiben.github.io/dfwmeetup_2014/#/

Using and Writing Jupyter Notebooks

Prof. Susanna Werth

Using and Writing Notebooks

Download (from Canvas) and Investigate the following Notebooks during or after Class:

- **Using Notebooks**: L01_tutorial_PythonFundamentals.ipynb

- **Writing Notebooks**: L01_reading_MarkdownAndMagic.ipynb

Prep for next Class

- Create a GitHub account: https://github.com
  - We will use GitHub, to make sure to have course content and your work available everywhere
  - Use any user name of your choice

- Optional Reading:
  “Interactive notebooks: Sharing the code”, Helen Shen. (2014). *Nature*, *515*, 151-152.

Supplement

Jupyter Notebooks Web Apps

Prof. Susanna Werth

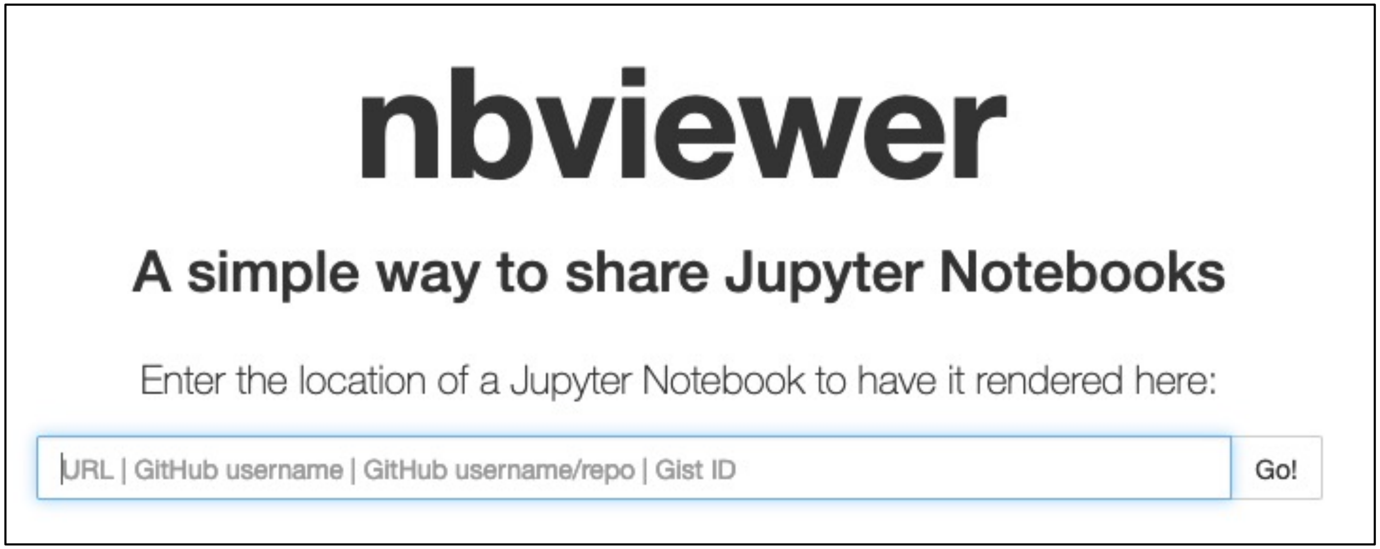
Webtools to work with Notebooks

Before we get to examples, here some options to run and write notebooks on the web (no software installation on your computer needed):

- nbviewer
- binder
- google colab
- ...

Jupyter NB Viewer

- **Purpose:**



nbviewer

A simple way to share Jupyter Notebooks

Enter the location of a Jupyter Notebook to have it rendered here:

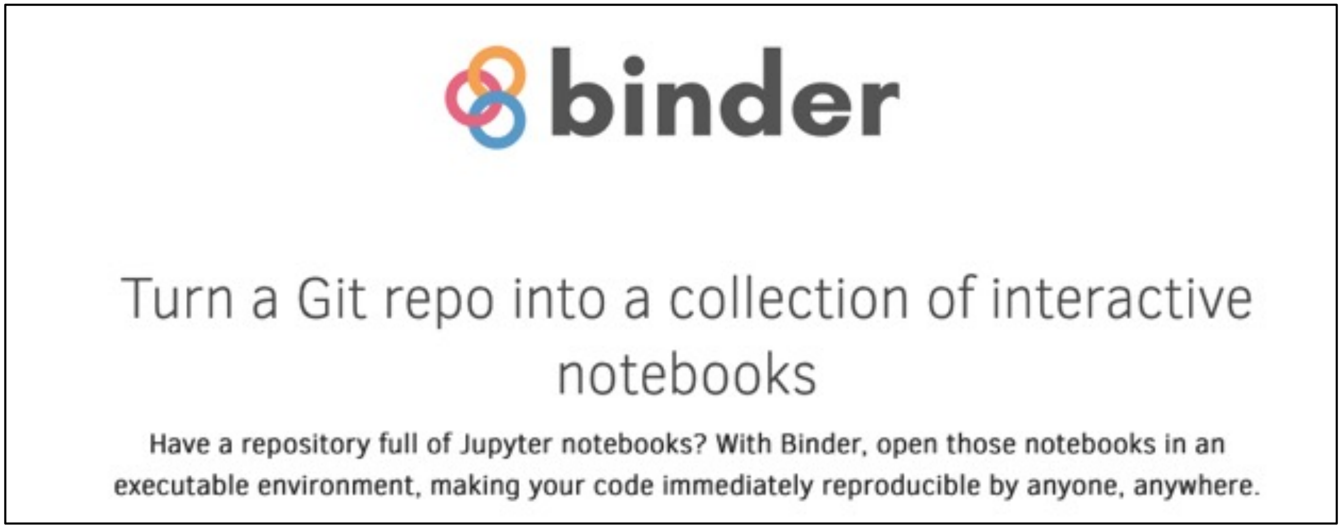
URL | GitHub username | GitHub username/repo | Gist ID

- **Main page**: nbviewer.jupyter.org
- **Purpose:** Read your notebook stored at a URL
- **GitHub** also allows you to view Jupyter notebooks (not all features might work).

Jupyter Binder Project

(for Jupyter Notebooks)

- **Purpose:**



binder

Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

- **Main page**: mybinder.org
- **Purpose**: Interactively work on your notebook stored at a URL
- Launches server with a jupyter notebook app
- Notebook environment without installation
- github.com/GeoPythonVT/JupyterWorkshop

Google Colab
(for Jupyter Notebooks)

- **Main page**: colab.research.google.com
- **Purpose:** Free Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive.
- Read, execute, write Jupyter Notebooks without software configuration
- Collaborative ability (like google-docs)
- Standard packages cannot be installed permanently (yet) and are not constant during the last year.
- Future unknown, but worth to keep observing
- Tutorial on Machine Learning Python packages