THE UNIVERSITY OF ARIZONA

BE/BAT 485/585

Remote Sensing Data and Methods

Lab - 1

Instructor: Kamel Didan[1,2]

Helpers: Dr. Armando Barreto[1,2]

Mr. Truman Combs[1,2]

[1]BE Dept., University of Arizona,

[2]VIP Lab.

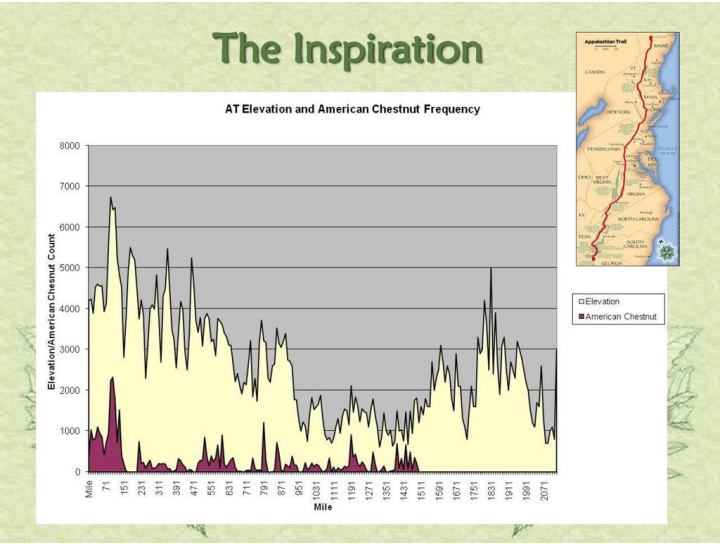
Scalars, Vectors, Matrices, and Tensors

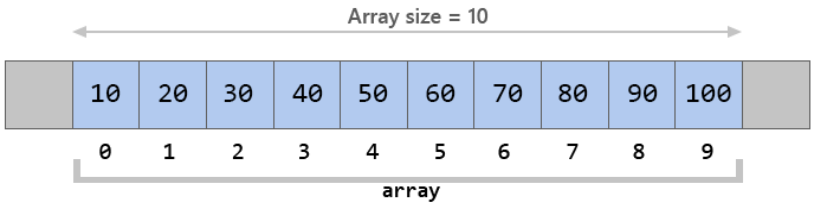
Click

- **Scalar**
  - A single quantity that has magnitude only
    - For example, $x \in R$ (real numbers) or $y \in N$ (Integer Numbers)
    - Example scalar quantities:
      - Length, Area, Speed, Volume, Temperature, Pressure, Energy, Work, Power, Entropy

Scalars, Vectors, Matrices, and Tensors

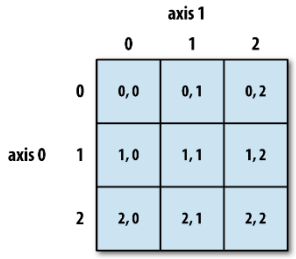
- **Vector**
  - A vector quantity has both magnitude and direction
    - Example: Displacement, Acceleration, Momentum, Force, Elec. Field, Magnetic field.
  - Also, an ordered/indexed list of numbers/scalars
    - For example, $x \in R^n$
    - Points in space where each element is a coordinate on a different axis
  - Example: Height or count along a hiking trail



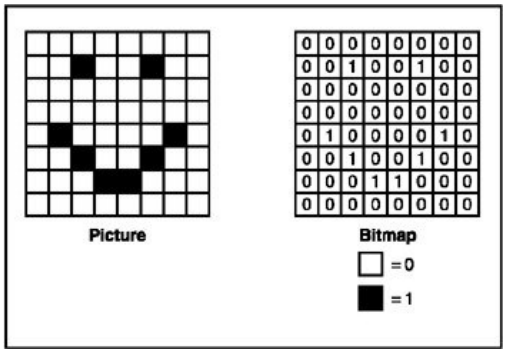


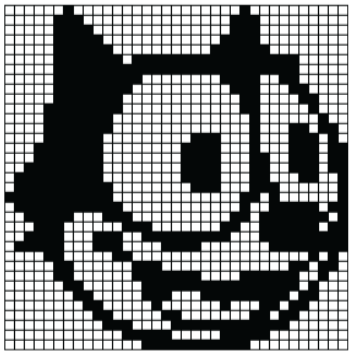
Scalars, Vectors, Matrices, and Tensors

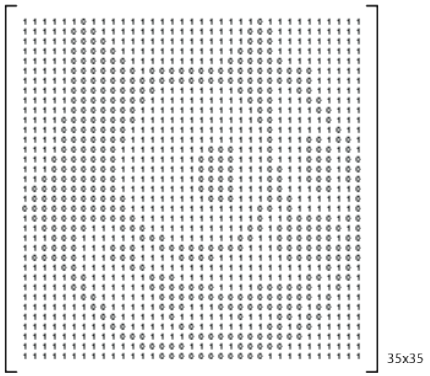
- **Matrix**: 2D Array of numbers
  - For example $X \in R^{m \times n}$



- Ex: The value of a pixel in a B/W image (or color)



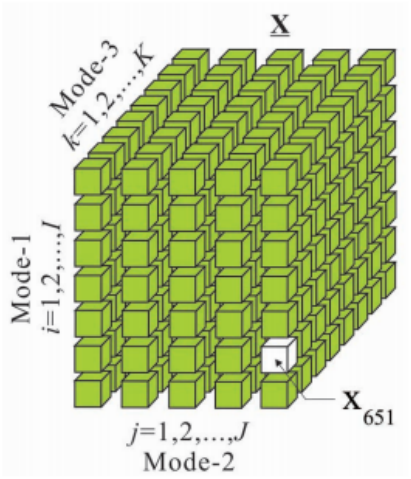




- Organized into Rows vs Columns (indices)
  - RxC, LxP, i x j, etc.
  - Helps address and navigate the content

Scalars, Vectors, Matrices, and Tensors

- **Tensor**: Arrays of numbers arranged on a regular grid with variables number of axes
  - Tensors are data containers and are critical and basic building blocks of modern data science & machine learning
  - A higher-order tensor can be interpreted as a multiway array, [...]
  - A tensor can be thought of as a multi-index numerical array, [...]



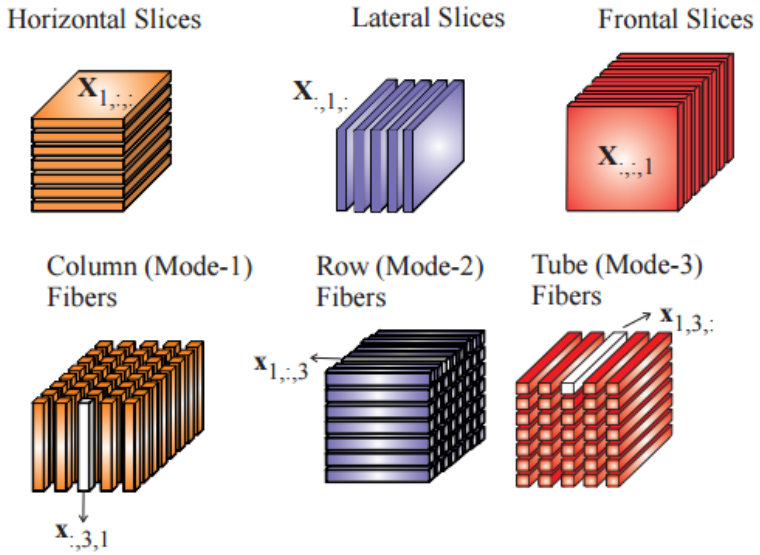
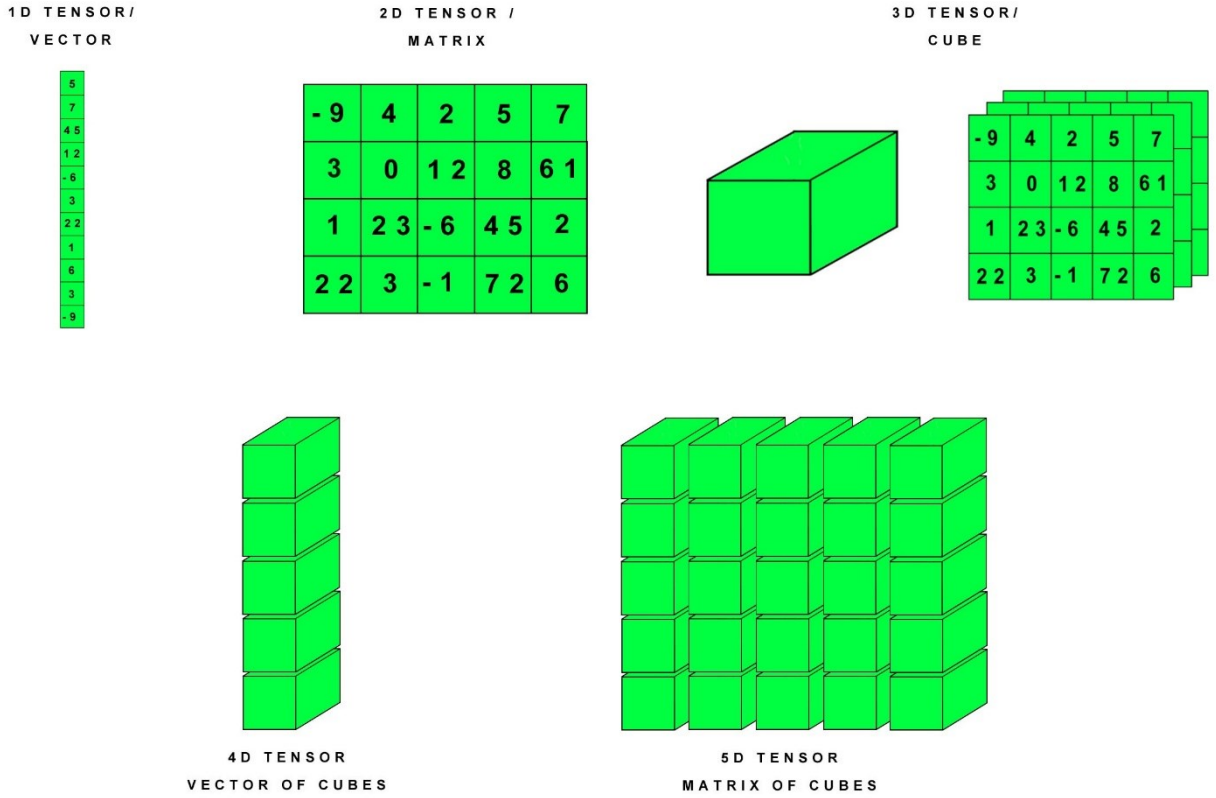


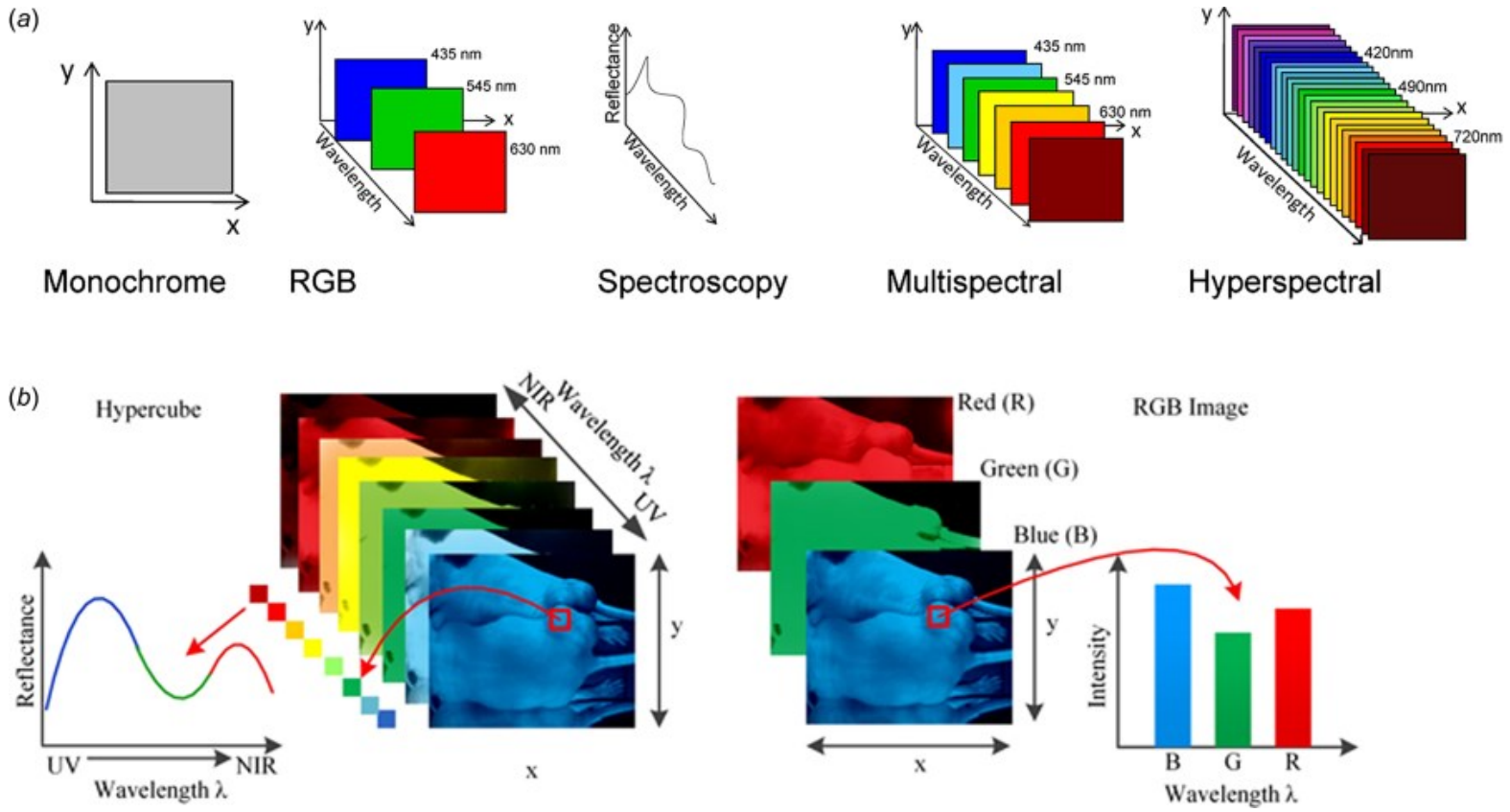
Figure 2: A 3rd-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$, with entries $x_{i,j,k} =$

But they are all Tensors at the end

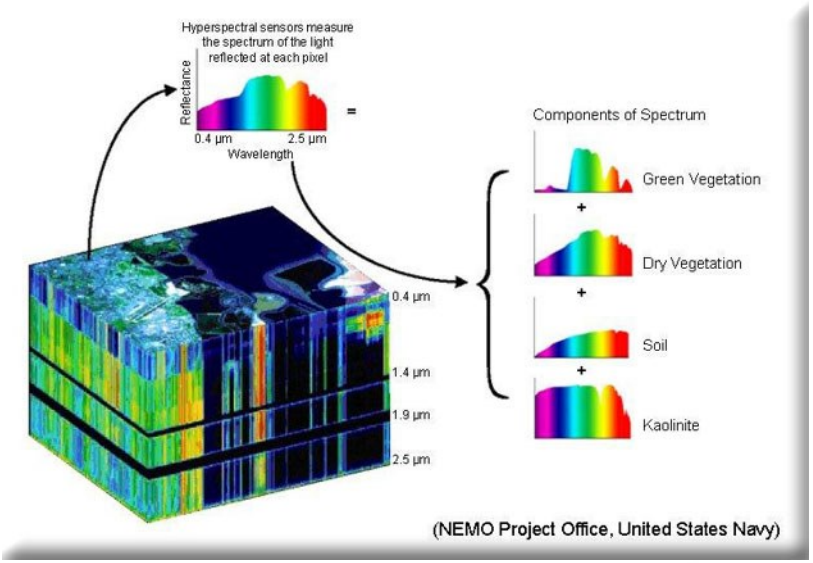
- *0*D, *1*D, *2*D, 3D, *n*D
- Containing scalars (sometimes other forms of data)

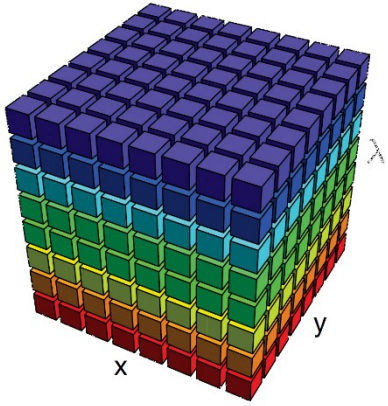


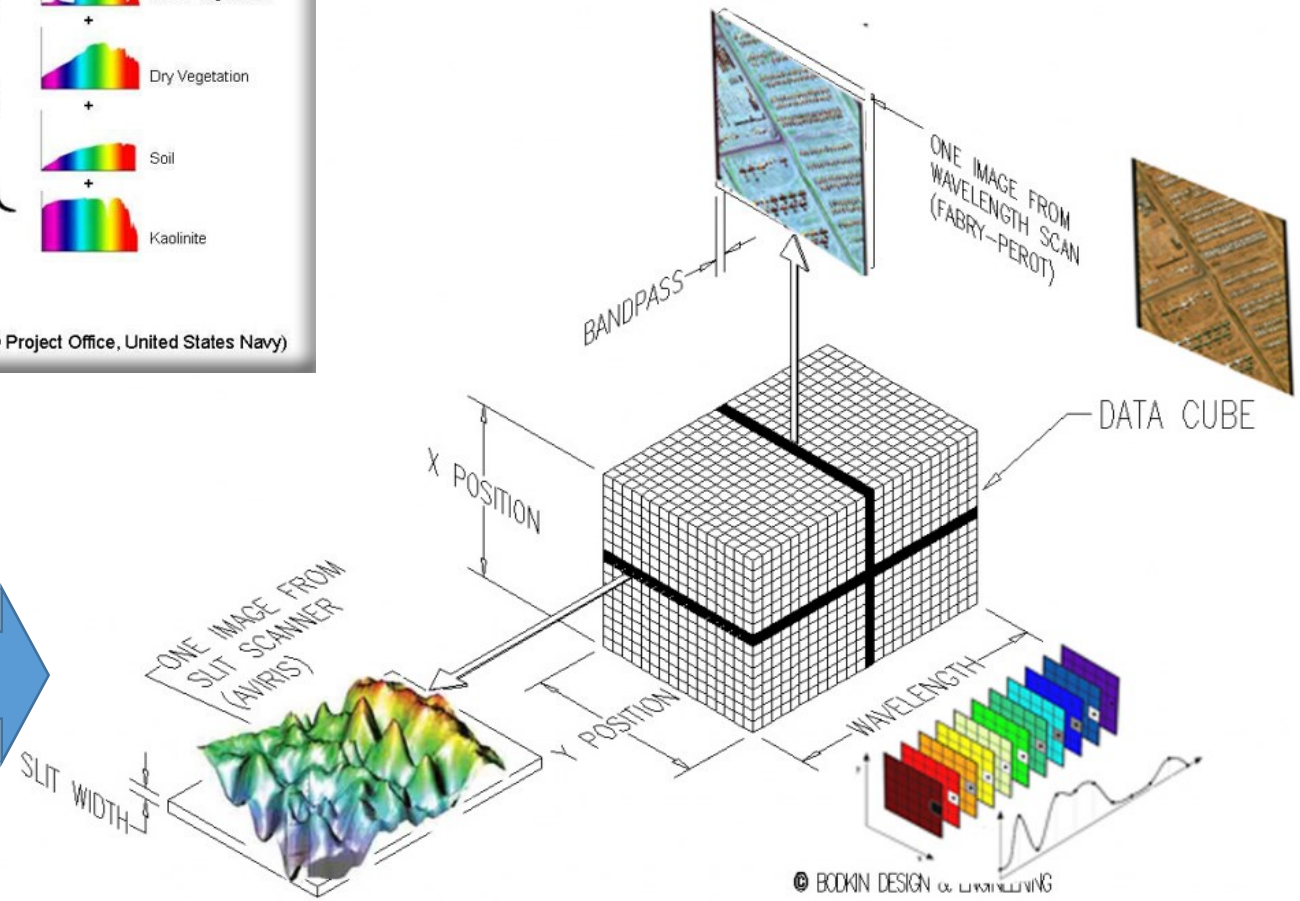
RS Data representation...become familiar with



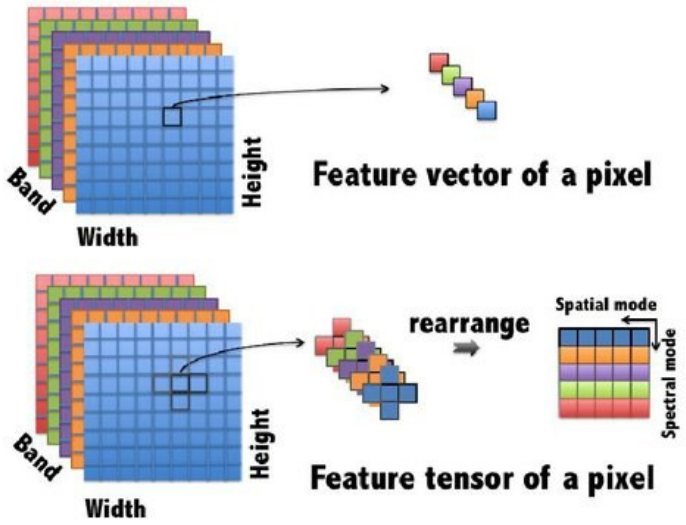
Become familiar with

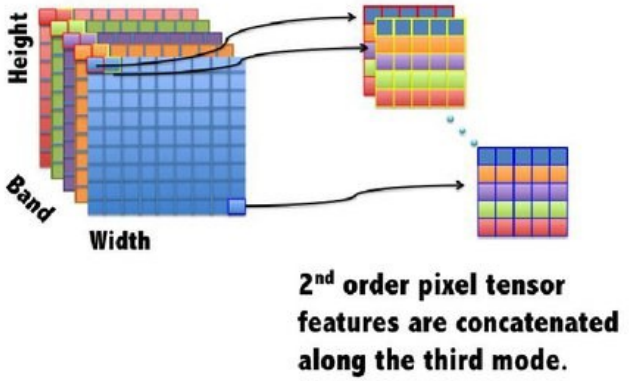






Become familiar with

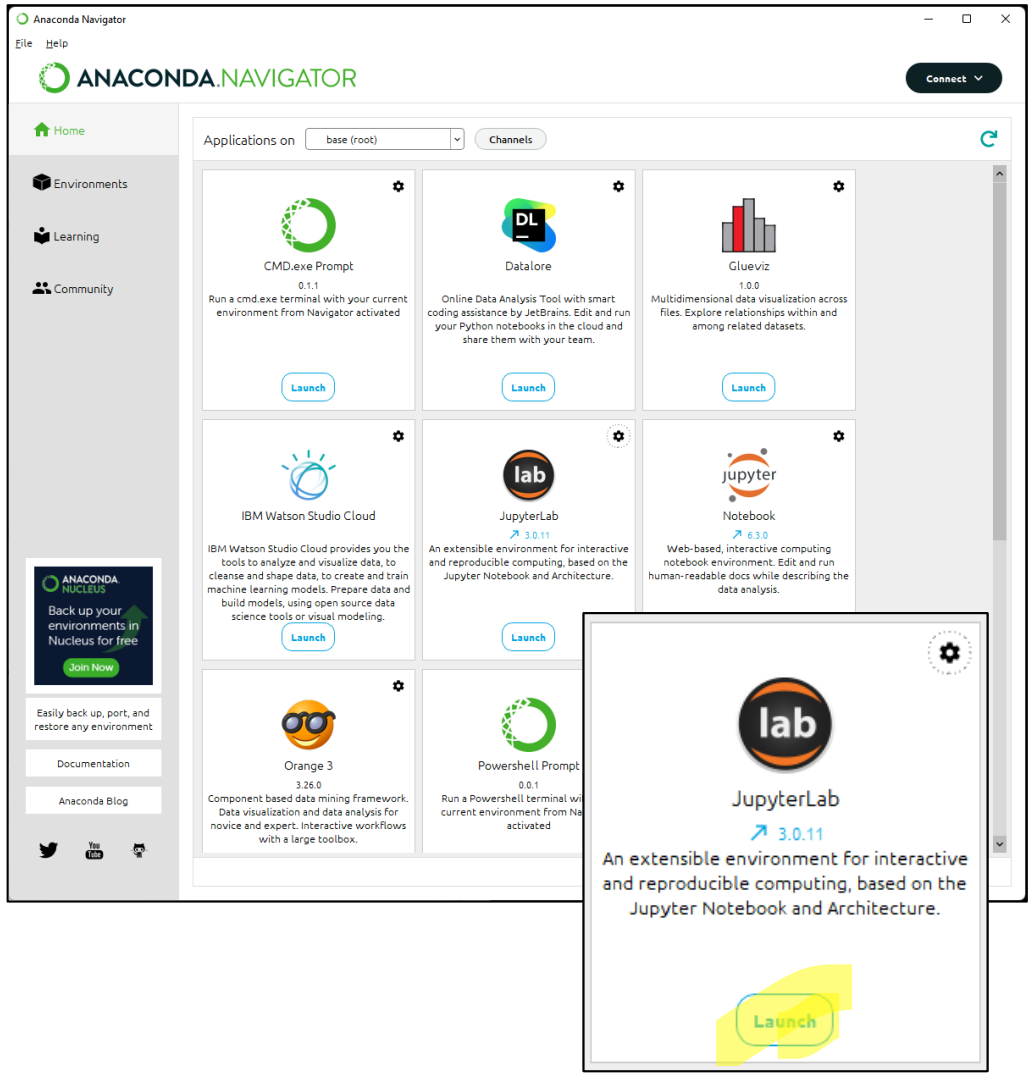




Programming with Python – Quick Intro

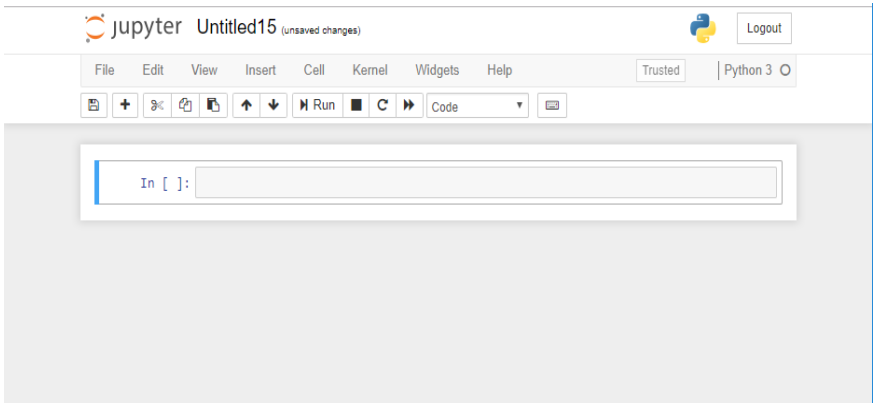
We will use Anaconda in this Course

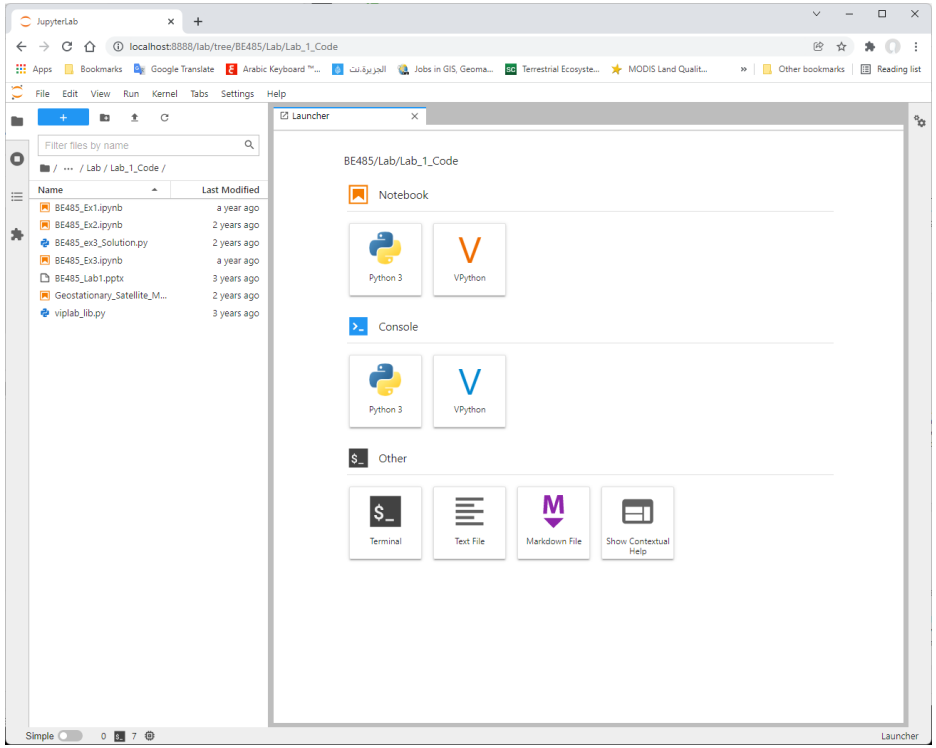
- Get and install the latest version of Anaconda from
  - www.anaconda.com
  - Follow instructions
- We will add libraries as needed
- It is advisable to set it only for yourself
- We will also use ***Jupyter Lab***
  - Like Notebook but more integrated with the browser



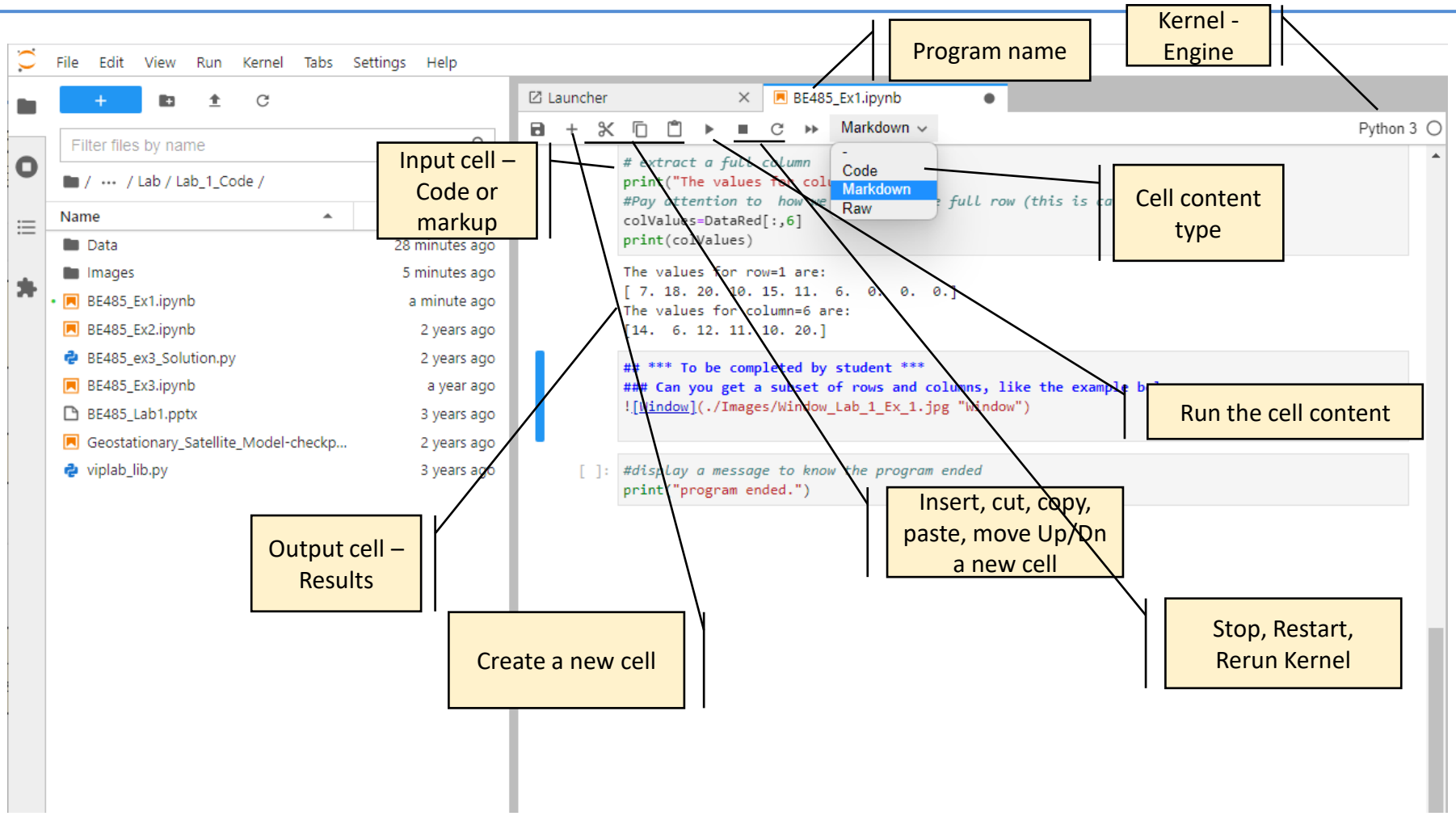
Intro to Jupyter Notebook and JupyterLab

- **Jupyter Notebook** is a web application that allows you to create and share interactive documents containing live code, equations, visualizations, and narrative text, effectively creating rich and live interactive reports.
  - **Jupyter Notebook** offers several different building blocks for interactive computing: the notebook, file browser, text editor, terminal, outputs, etc.
- **JupyterLab** is an integrated rich "data science" UI (very similar to Notebook) but offers more.
  - It is a natural evolution of the classic notebook and provides a more flexible and powerful way of working with the same building blocks found in the notebook.
  - **Jupyterlab** is 'almost' the same as **Jupyter notebook** but more flexible, more powerful, has more tools, and more integrated with the browser

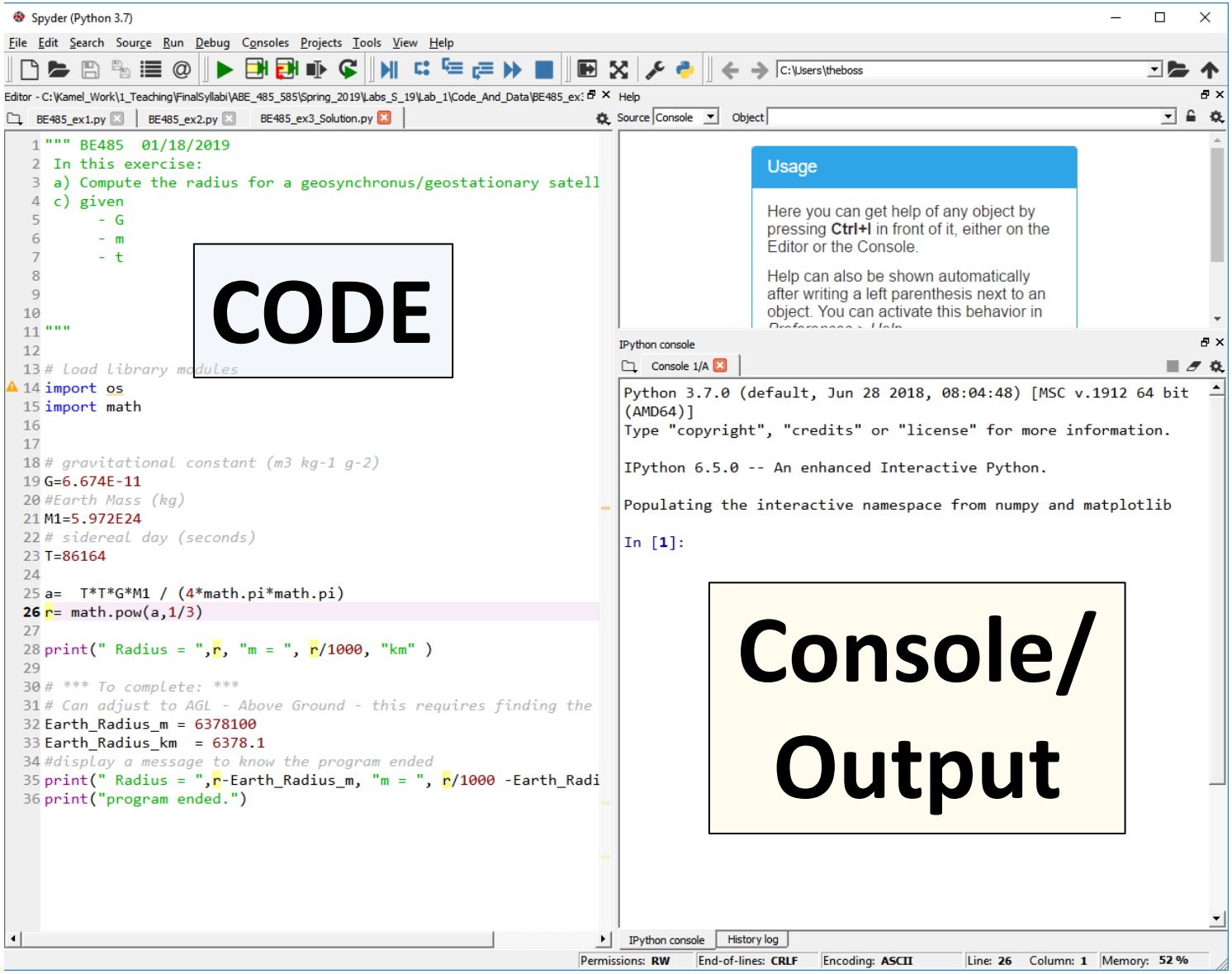




Jupyterlab - View



Then there is Spyder and PyCharm IDEs

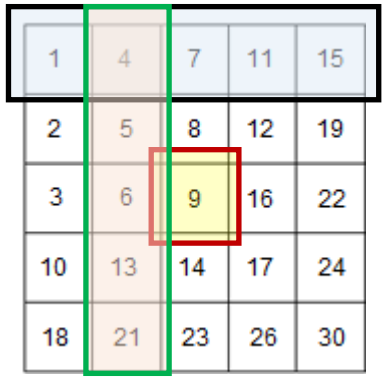


Which one to use

- We will use **JupyterLab** but feel free to use whatever suits you
  - And you may have to be on your own at times
  - You get the same functionality and more
- Standardize our Lab work/HW and makes it easier to grade, debug, and help
  - You will always turn in your notebook (*.ipynb)
  - And we will run and see the results
  - Learn to annotate and clean your work
  - Learn to create nice scientific reports

Exercise #1: Arrays and Data Access

- Arrays and datasets
  - Create a 2D array
  - Access a single pixel value
  - Retrieve a row of data
  - Retrieve a column of data



  Note: Indexes are Zero-based, so first row or column is always **zero**

- **Homework**:
  - Change the code to extract a 'subset/group' of values
  - Create a new 2D array that contains the values from row 2-4 and columns 3-6
  - Print this new array to the screen

Instructions:

- Download from D2L files:
  - BE485_UofACampus.xlsx
  - BE485_Ex1.ipynb

Exercise #2: Arrays and Data Access

- Reading data and creating Images from an Excel Table
  - Read an Excel file
  - Access the data in the Excel sheet(s)
  - Display single band/channel Images (image are 2D arrays of scalars)
  - Display an RGB Image (combine the 3 bands). That is how images are created

- Homework:
  - Using code from **Ex1**, display a window/subset that represents 'Old Main' building as an RGB image
  - You will need to define/find the extent of Old_Main
    - You can guess by estimating the corners' locations, then fine-tune it

Instructions:

- Download from D2L files:
  - BE485_UofACampus.xlsx
  - viplab_lib.py [Library]
  - BE485_Ex2.ipynb

RGB Images

RGB Image
Number of rows = 751
Number of columns = 1151