REU Python 3 Workshop

Brought to you by APS

July 23rd, 2020

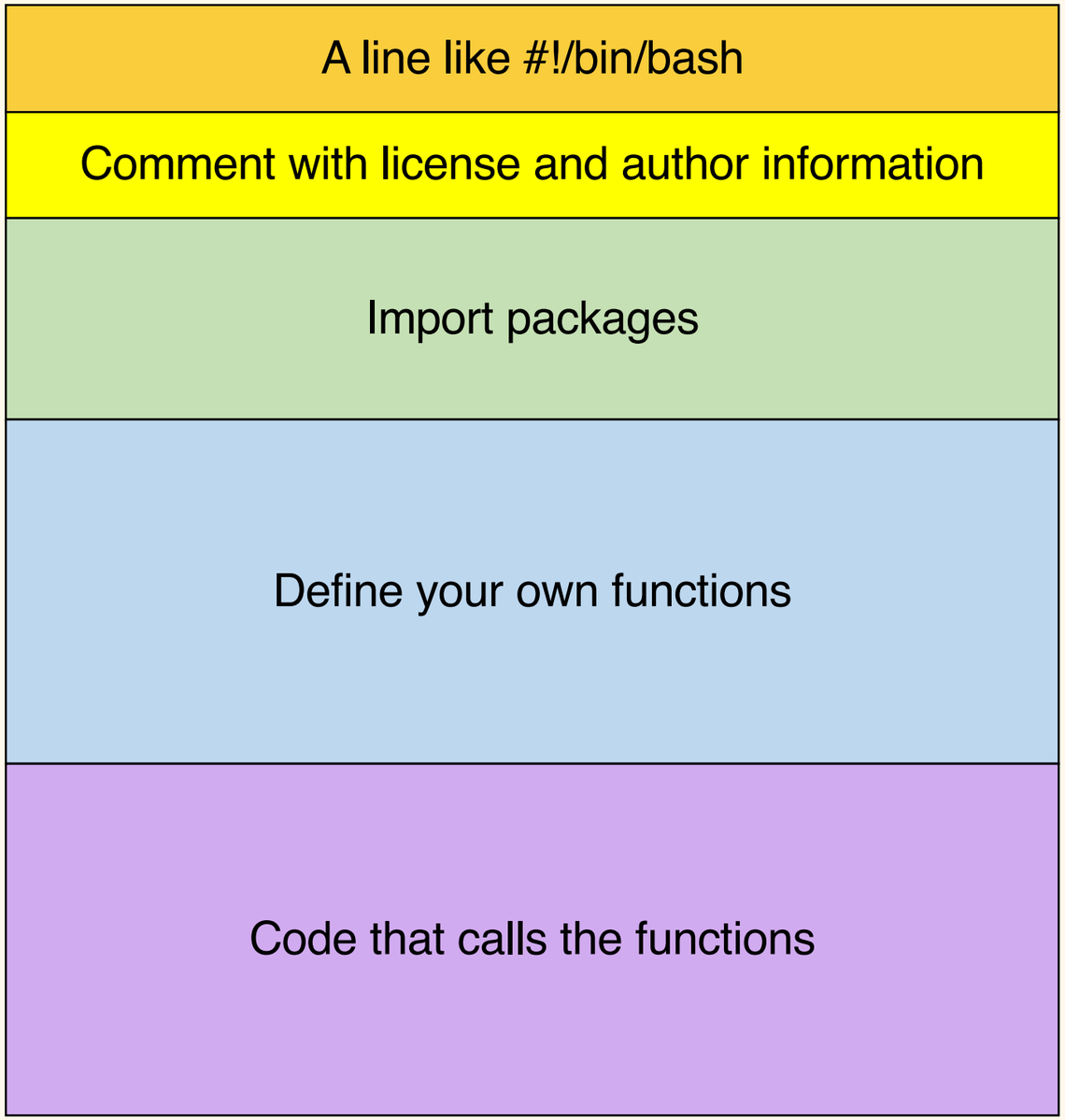
Today’s Agenda

12:00 - 12:10 PM (ET)	*Introductions and Housekeeping*
12:10 - 1:00 PM (ET)	*Python 3 basics (Syntax, data structures, loops, and user-defined functions)*
1:00 - 1:10 PM (ET)	*[Break]*
1:10 - 1:25 PM (ET)	*Matplotlib Example with Breakout Room Work*
1:25 - 1:40 PM (ET)	*Scipy and Pandas Example with Breakout Room Work*
1:40 - 1:50 PM (ET)	*FITS Handling Example with Breakout Room Work*
1:50 - 2:00 PM (ET)	*Wrap Up*

What’s in the Folder?

1_syntax.ipynb

Python Code Structure



 # short comments sprinkled throughout

‘’’
Some really long comments too,
which are enclosed by triple quotes
‘’’

print(‘Print functions whenever you want
to see code output’)

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created for APS Summer REU Python 3 Workshop

@author: victoriacatlett
"""

################################################################################
############## Import what you need from packages at the top ###################
################################################################################

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

################################################################################
################### Define a function we'll need later #########################
################################################################################

def mySin(x,a,b):
    y = a*np.sin(b*x)
    return y


################################################################################
############## TASK 1: Read in the Excel file OR the CSV file ##################
################################################################################

path_to_xlsx = '../../files/pandas.xlsx'
#path_to_csv = '../../files/pandas.csv'

data = # YOUR CODE HERE: Read in the data
x =    # YOUR CODE HERE: Get the x data
y =    # YOUR CODE HERE: Get the y data


################################################################################
################## TASK 2: Plot data (BONUS: Plot fit sine) ####################
################################################################################

# This creates two places to plot (ax[0] and ax[1])
# on a single figure called "fig"
fig, ax = plt.subplots(nrows=2, ncols=1)
plt.subplots_adjust(hspace=0.5)

# Set titles for the two plots
ax[0].set_title('Original Data')
ax[1].set_title('Fit Sine Curve')
```

2_variables.ipynb

2_variables: Variables

Variables store values for future reference.

Variable Type	Definition	Examples
boolean	True or False	True, False
integer	A whole number *without* a decimal point	... -2, -1, 0, 1, 2 ...
float	Numbers *with* a decimal point	3.14, 100.0001, 7.
string	collections of characters surrounded by quotation marks	'Hello!', "1234", "etc."

Ex) Say you have a code that tells you how old you'll be in certain years. It'll need your current age.
To create a variable called my_age which stores your age, just type the line

my_age = 20

Python knows this is an integer, and now you don't have to type 20 everywhere!

2_variables: Data Structures

Data Structures can hold more than one value at once

Variable Type	Definition	Examples
list	A list of variables	my_list = [1, ‘two’, 3.0]
array	A matrix. (Essentially a list of lists.)	my_array = [[1,2,3],[4,5,6],[7,8,9]] = $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$
dictionary	A set of “keys” and their associated “values” (like a word and its definitions). Keys must be integers or strings.	my_dictionary = {'Year':2019, 'School':'UTD’, ‘Some_Numbers':[1,2,3]}

my_list[0] = 1

my_array[1][2] = 6

my_dictionary[‘School’] = ‘UTD’

3_functions.ipynb

3_functions: Built-In Functions

Function	Output
print(x)	Displays string x in the console
type(x)	The variable type of x
abs(x)	The absolute value of a number x
range(n)	The sequence {0, 1, … , n-1}
list(range(n))	The list [0, 1, … , n-1]

This is a short list of common functions you may encounter.
See online Python documentation for the full list.

3_functions: Boolean Operators

Boolean operators create an expression that is either **True** or **False**.

Python Operator	Meaning
and	and
or	or
not	not

That’s easy to remember! Here are some examples:

True **or** False = True
True **and** False = False

3_functions: Comparison Operators

Python Operator	Meaning
<	less than
< =	less than or equal to
>	greater than
> =	greater than or equal to
= =	is equal to
! =	is not equal to

7 = = 8 is False

3 < 4 is True

3_functions: Loops

For Loop: Repeats a process a specific number of times

```
forSum = 0

for i in range(11):
    forSum = forSum + i
```

While Loop: Repeats a process over and over until the given condition is false

```
whileSum = 0
x = 0

while x < 11:
    whileSum = whileSum + x
    x = x + 1
```

Both of these loops evaluate the sum of 0-10

Note: range(n) returns the list [0, 1, …, n-1]

3_functions: User-Defined Functions

General Format

```
def functionName(inputs):
    output = [do something to the inputs]
    return output

specificOutput = functionName(specificInputs)
```

Example: y = mx+b

```
def line(x, m, b):
    y = m * x + b
    return y

y0 = line(0, 3, 2)
```

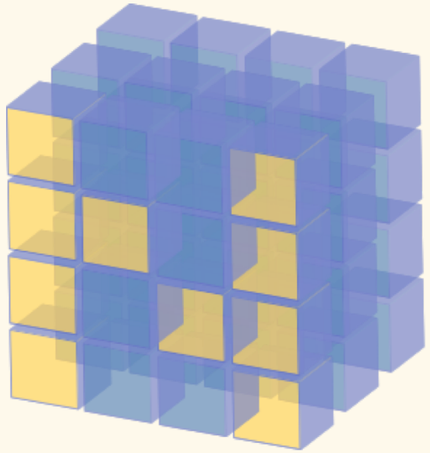
4_libraries.ipynb

Mathematics Packages

NumPy and **SciPy** are packages for mathematics and computation.

You will almost *always* need NumPy, as it improves and adds to the built-in Python math.

SciPy contains many common scientific constants and functions.



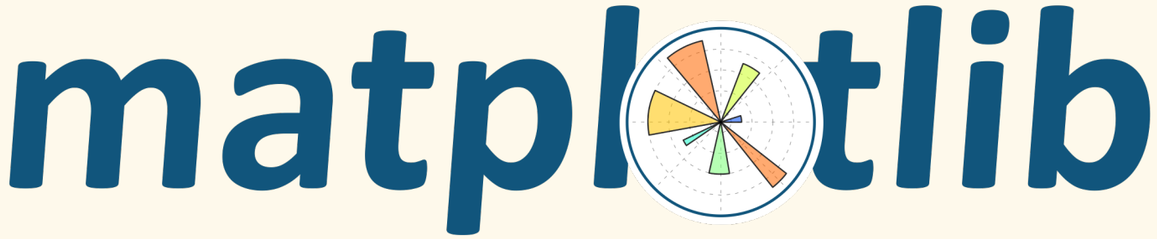
NumPy

SciPy

Plotting Packages

Matplotlib and **Seaborn** are two of the most common (and powerful) data visualization libraries.

We will only work with Matplotlib today.



Pandas

Pandas is a powerful data analysis library which works with **Data Frames**.

Data Frames are data in a **table format**, such as Excel spreadsheets or Comma Separated Values (CSV) files.

	A	B	C
1	x	y	
2	0	0.04746958	
3	0.01	0.23932802	
4	0.02	0.15274071	
5	0.03	0.16962677	
6	0.04	0.30859576	
7	0.05	0.38257724	
8	0.06	0.3292564	
9	0.07	0.38616885	
10	0.08	0.38549168	
11	0.09	0.43197485	
12	0.1	0.51758194	

```
index,x,y
0,0.0,0.04659049277507954
1,0.01,0.06642411315729269
2,0.02,0.13102361953897132
3,0.03,0.22867759086606038
4,0.04,0.3468186080452961
5,0.05,0.36043418819287615
6,0.06,0.3438464959254184
7,0.07,0.36819030282979937
8,0.08,0.38670691500471466
9,0.09,0.42290156458823214
10,0.1,0.48125929691252844
```

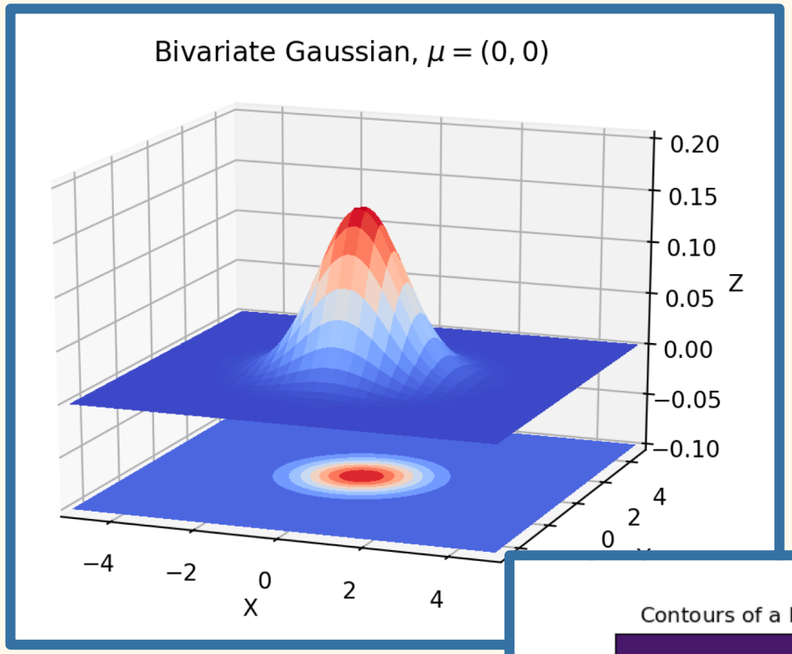
[Break]

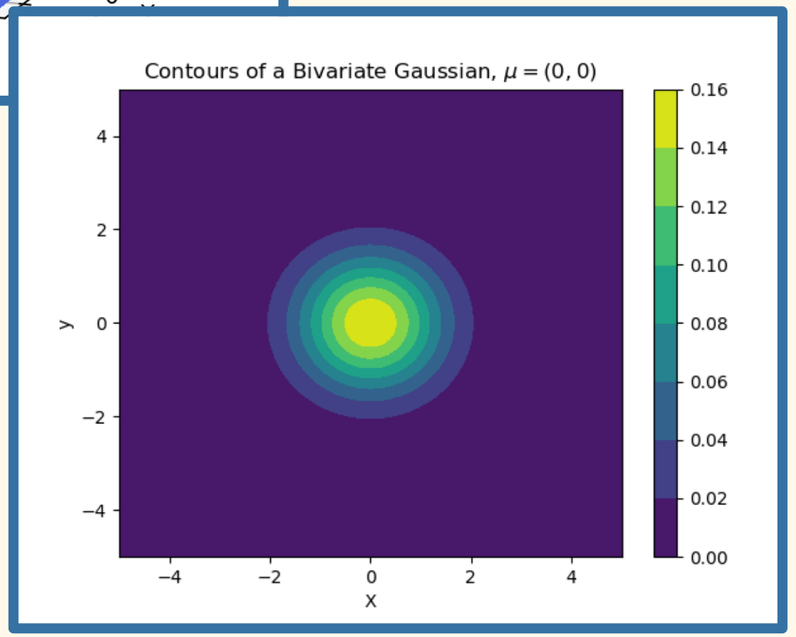
Part 2:
Breakout Room Python Examples

extra_plot3d: Create 3D Plots

This example will guide you through:

1) Plotting a bivariate Gaussian
2) Plotting its x-y projection in 3D
3) Plotting its x-y projection in 2D



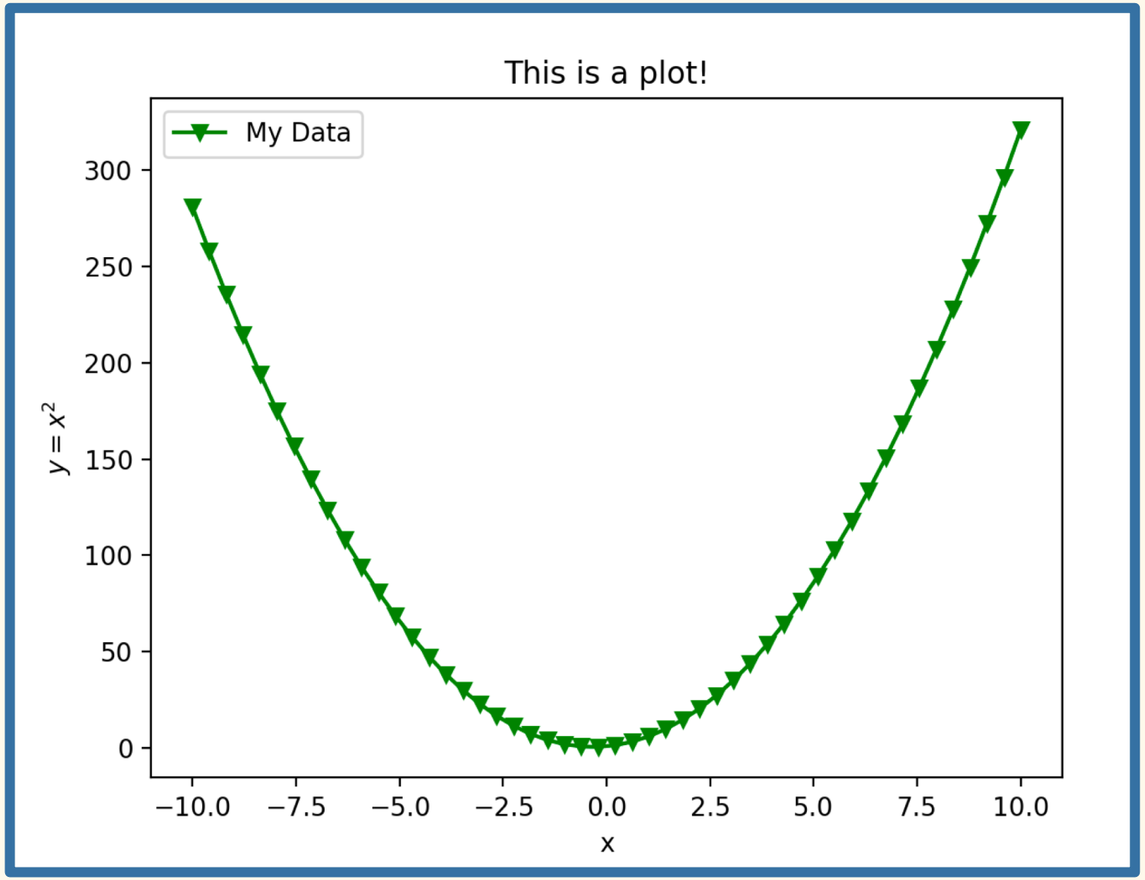


Challenge 1:
1_matplotlib.ipynb

1_matplotlib: Plotting with Matplotlib

In this example, you will:

1) Plot the function
   $y = 3x^2 + 2x + 1$
2) Add an equation to an axis label
3) Save the figure as a PNG



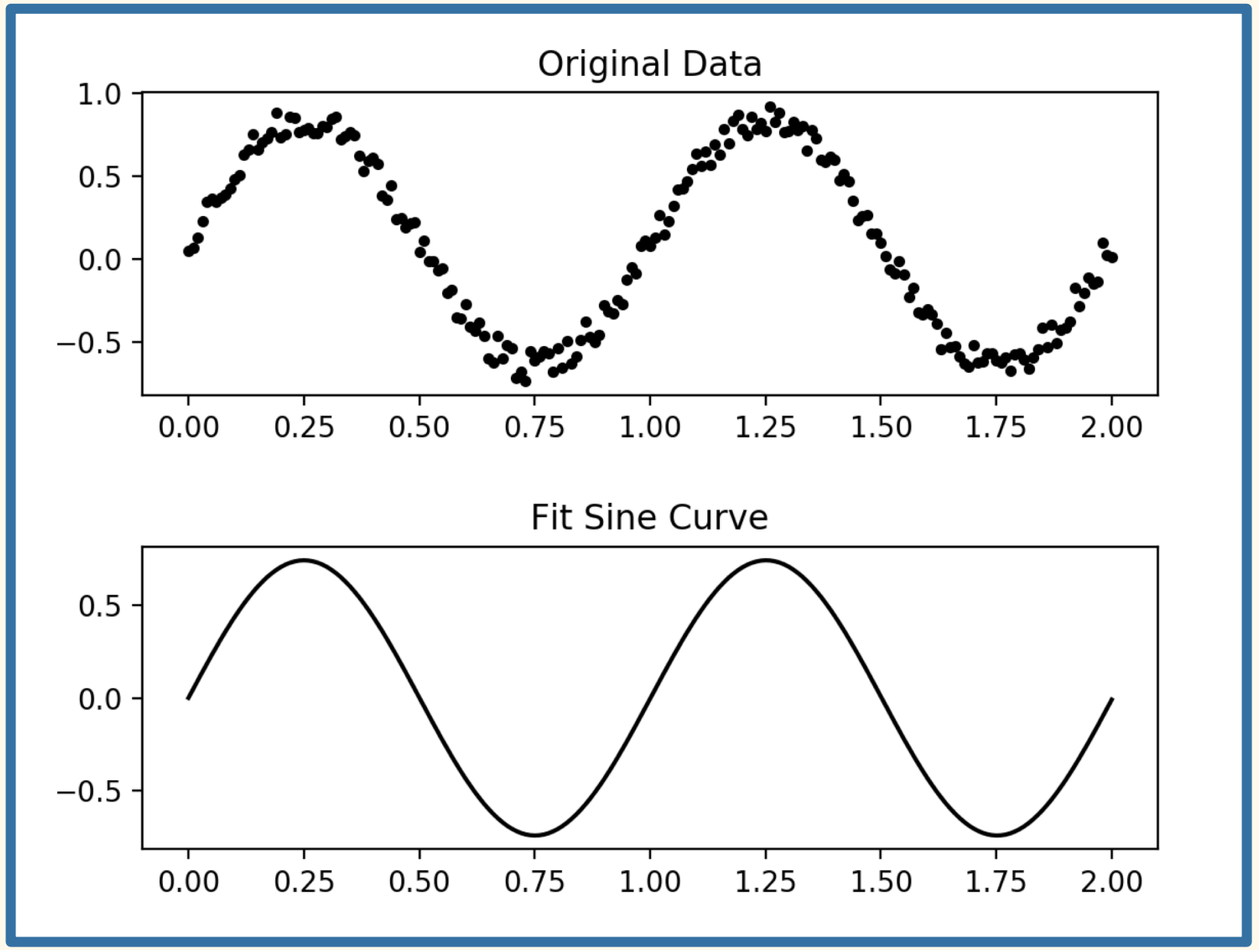
Challenge 2:

2_scipy_pandas.ipynb

2_scipy_pandas

In this example, you will:

1) Pull data from an Excel sheet (or CSV)
2) Fit a sine curve to the data
3) Plot everything
4) Save the data to a new file



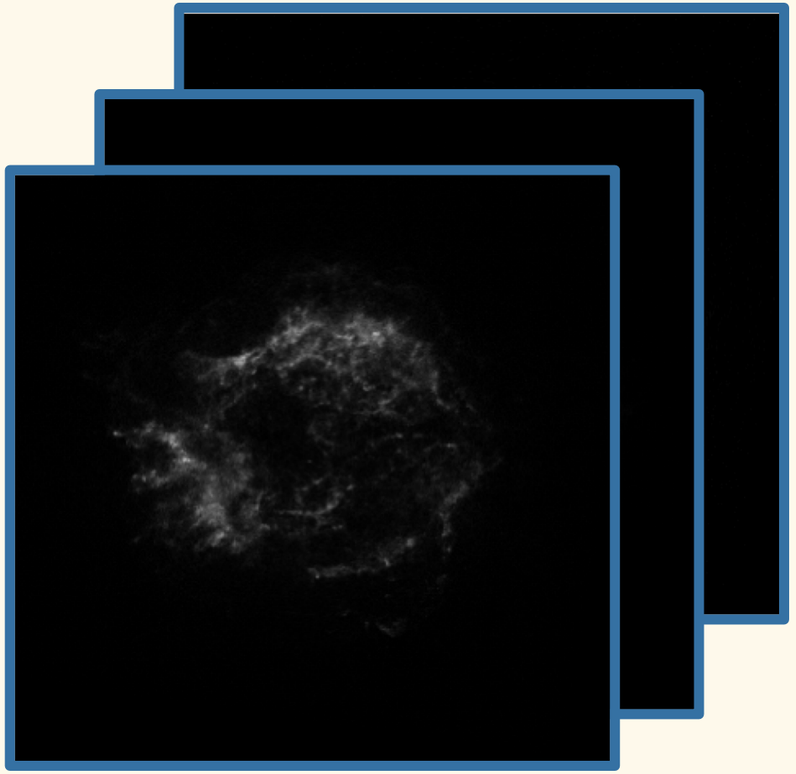
Challenge 3:
3_fits.ipynb

Flexible Image Transport System (FITS)

FITS is the standard data type for astronomical images, and it can used in a variety of other applications.

FITS files contain one or more Header Data Units (HDUs), which each contain an image (the data) and keyword descriptions (the header).

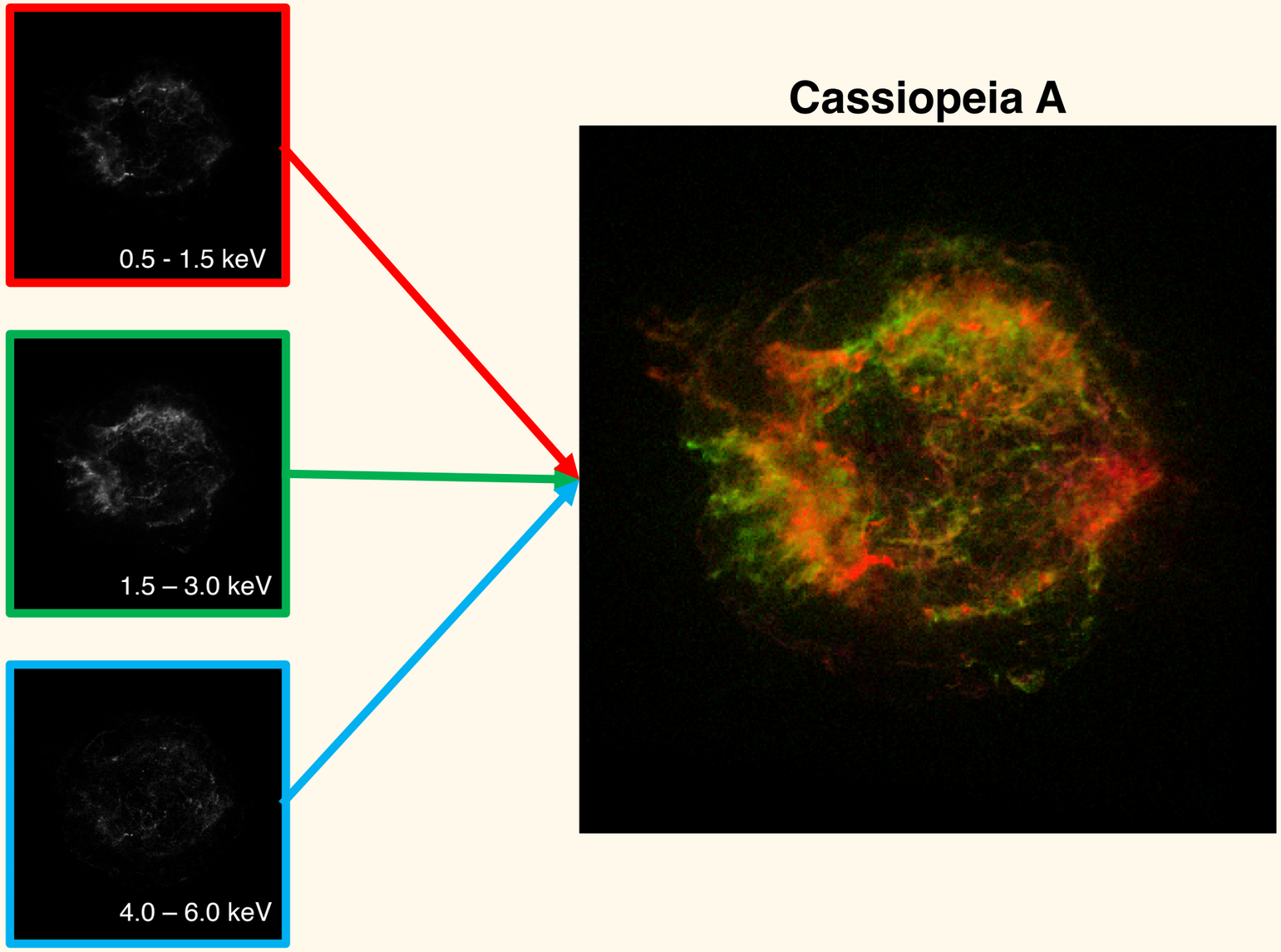
Essentially, the files are stacks of images which each have some text to describe them.



3_fits: Use Astropy to work with FITS

In this example, you will:

1) Open a FITS file with 3 HDUs
2) Read header information
3) Turn the 3 images into an RGB image

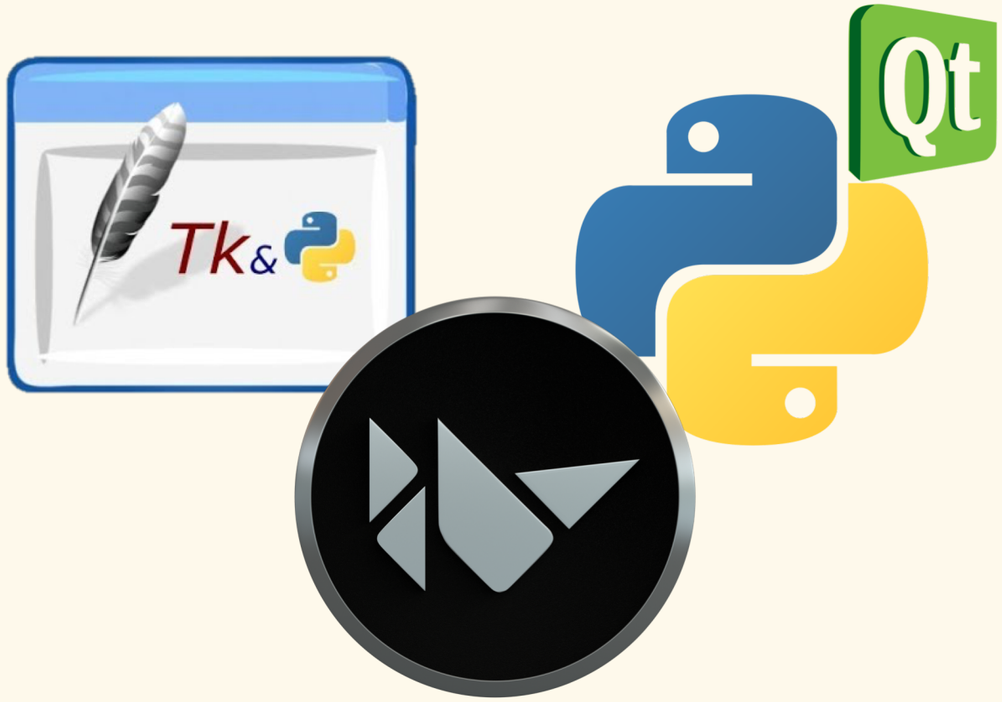


Data Source: Chandra Photo Album

Wrap Up:

Other Python Packages

Other Potentially Useful Packages

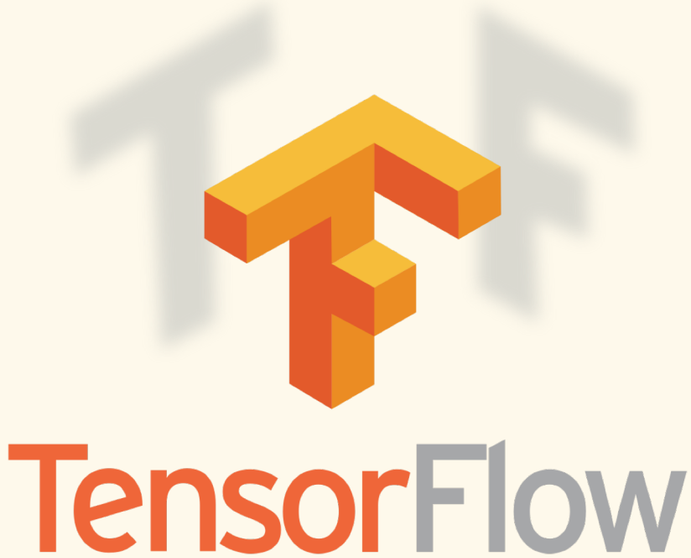


TkInter, Kivy, and PyQt will help you create cross-platform desktop applications

OpenCV is used for image recognition and processing, such as facial recognition or object detection.

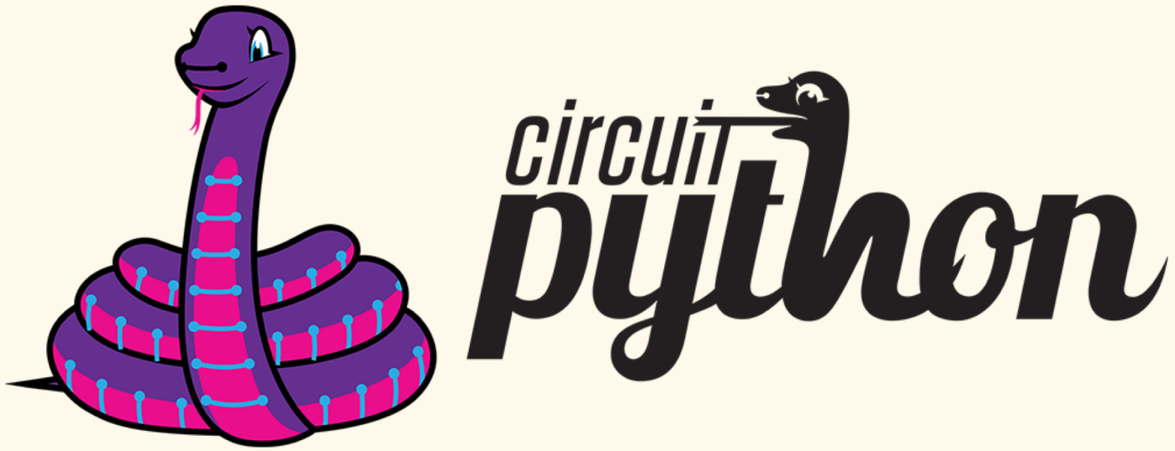
Other Potentially Useful Packages

Scikits are toolkit extensions to **Scipy**. There are all sorts of them, one of the most notable being **Scikit-learn**, which is commonly used for machine learning.



TensorFlow is another Python package for machine learning. It is especially useful for deep learning and making neural networks.

Fun Packages



Twilio allows you to send and receive phone calls and text messages via your program.

CircuitPython is almost exactly the same as Python, but it has support for hardware. It is produced by Adafruit, which makes compatible microcontrollers.

Thank you for joining this workshop!

If you want to reach out later, my email is
vec170000@utdallas.edu