Python DeCal

Week 4

Announcements

- 2nd Hw was due just now!
- Office Hour
  - Thanks to those who showed up!!
- Attendance!
  - https://forms.gle/hZQCUHm1p7uCN3Ex5

Recap

- What is recursion?
- How are dictionaries different from lists?

How to make our code more POWERFUL?

- `import` statements:
    - math, random, numpy…
- You can import packages and libraries so that you don't have to write your own functions
    - How would you calculate the median of a list of a 10 unordered numbers?

How do you import?

- `import package_name as alias:`

- For example, let’s talk about NUMPY today!
  - `import numpy as np`
- Generally put at the very beginning of your code

Numpy Arrays (1D)

- The most used tool in science

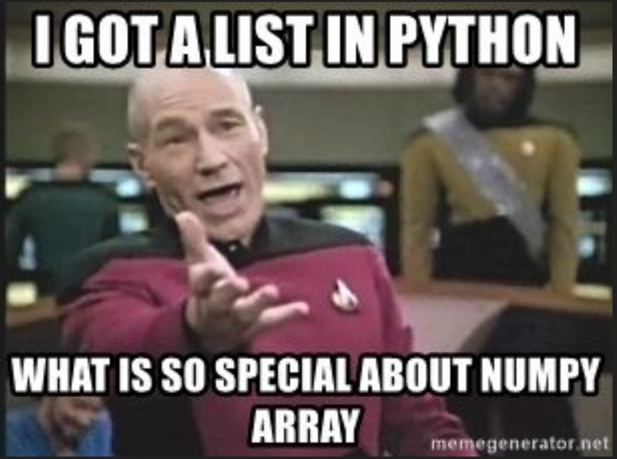
- This is a list

```
l = [0, 1, 2, 3, 4, 5, 6, 7]
```

- This is a numpy array

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7])
```

- Same indexing rule as lists



IT IS SUPERIOR!!!!!!!!

Some review: What do the following commands do?

```
list_1 = [0, 1, 2, 3, 4, 5, 6, 7]

list_2 = [0, 1, 2, 3, 4, 5, 6, 7]



3*list1 → ???

list_1 + list_2 → ??????
```

IT IS SUPERIOR!!!!!!!!

HOWEVER, THIS IS WHAT NUMPY ARRAY DOES!

```
    arr1 = np.array([0, 1, 2, 3, 4, 5, 6, 7])

    arr2 = np.array([0, 1, 2, 3, 4, 5, 6, 7])
```

Numpy arrays operate element-wise!

```
    3*arr1 → np.array([0, 3, 6, 9, 12, 15, 18, 21])

    arr1 + arr2 → np.array([0, 2, 4, 6, 8, 10, 12, 14])
```

Notice that the size of the two arrays have to be the same

What else can I do with 1D arrays?

Calculating the mean / average value:

```
sum(your_list)/len(your_list)

np.mean(your_array)
```

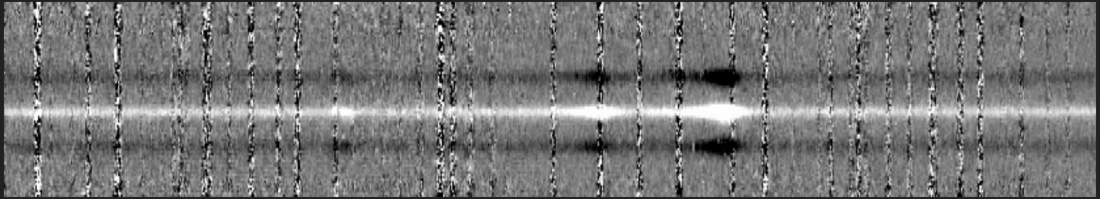
Calculating standard deviation (San Diego!!!)

```
np.std(your_arr)     np.percentile(your_arr, percentile)
```

Calculating median value:

```
np.median(your_arr)
```

2D Array!



You can think of it as a matrix, sometimes a table

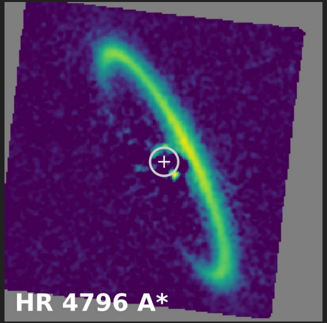
```
    np.array([[1, 2],
              [2, 1]]
```

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

You can store different information in a 2D-array

```
    np.array([[x1, x2, x3, x4, x5],
              [y1, y2, y3, y4, y5]]
```

Many astronomical datasets are 2D arrays! (images, spectra)



Slicing & Indexing (you'll use this almost everyday)

```
arr = np.array([[1, 2, 3],
                [2, 1, 3],
                [3, 2, 1]])
```

$$\begin{array}{cc} \text{Axis 0} & \begin{bmatrix}1 & 2 & 3\\ 2 & 1 & 3\\ 3 & 2 & 1\end{bmatrix} \\ & \text{Axis 1}\end{array}$$

Of course, we can have more axes.

Slicing & Indexing (you'll use this almost everyday)

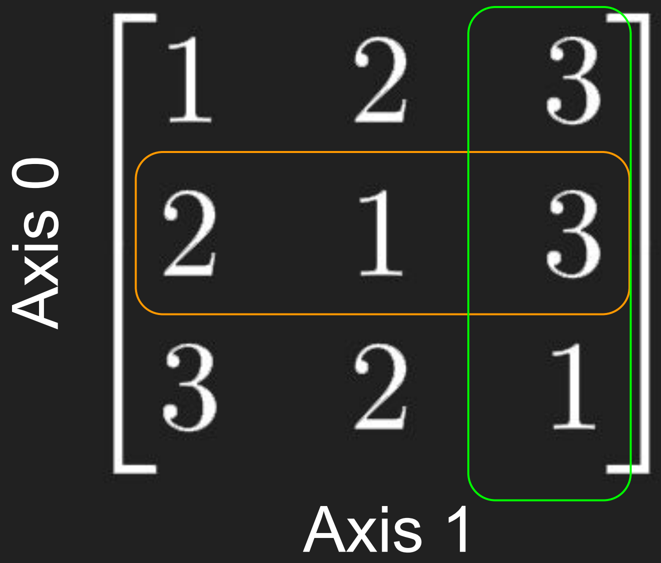
```
>>> arr[1, 2]
    3
```

Index along axis 1

Index along axis 0

Everything along an axis

```
>>> arr[:, 2]
    np.array([3, 3, 1])
```


$$\begin{array}{c} \\ \text{Axis 0} \\ \\ \end{array} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix}$$

Axis 1

Now I want the middle row….

“Axes-wise Operations”

```
>>> np.mean(arr, axis=0)
    np.array([2,1.67,2.33])


>>> np.median(arr,axis=1)
    np.array([2, 2, 2])
```

$$\begin{array}{cc} \text{Axis 0} & \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix} \\ & \text{Axis 1} \end{array}$$

Extra Material

Numpy stats functions

Sometimes we want to directly know some information about the array we have.

- Order Statistics
- Averages and Variances

The Basics

- `numpy.max()`

- `numpy.min()`

Assumes a 1D Array or a List

Order Statistics for Higher Dimensional Arrays

These functions will give us some insight on the range and distribution of the data.

```
numpy.amin outputs the minimum along a specified axis

numpy.amax outputs the maximum along a specified axis
```

Order Statistics Demo

```
numpy.amin outputs the minimum along a specified axis

a = np.array([[1, 2],
              [2, 1]]

np.amin(a)

np.amin(a, axis=0)
```

Order Statistics Demo

```
numpy.amax outputs the maximum along a specified axis

a = np.array([[1, 2],

              [2, 1]]

np.amax(a)

np.amax(a, axis=0)
```

Averages and Variances

`numpy.median`

`numpy.mean (arithmetic mean)`

`numpy.std`

`numpy.var`

Correlating

These functions will give us some insight on the correlations

```
numpy.amin

numpy.amax
```

```
Do you want the correlations?

Idk what they are but they are on the numpy stat functions
```

Histograms

These functions will compute the histogram of a given set of data

```
numpy.histogram

numpy.histogram2d

numpy.histogramdd

numpy.bincount

numpy.histogram_bin_edges

numpy.digitize
```