YData: Introduction to Data Science

Class 06: Array computation continued…

Overview

Brief review of:

- NumPy arrays
- Numerical array computations

Higher dimensional numerical arrays

Image manipulation

If there is time:

- Functions!

Questions?

Review of array computations

Review of NumPy arrays and functions

Processing data that is all of the same type can be more efficient than processing data of mixed types.

The *NumPy package* stores and process data that is all of the same type using ***ndarray*** and contains functions that operate efficiently on these arrays.

Review: ndarrays

```
import numpy as np

my_array = np.array([1, 2, 3])   # creating an ndarray from a list
my_array[0]                      # accessing the 0th element of the ndarray

my_array.dtype                   # get the type of elements stored in the array
my_array.shape                   # get the dimension of the array
my_array.astype('str')           # convert the numbers to strings

sequential_nums = np.arrange(1, 10)    # creates numbers 1 to 9
```

Review: NumPy functions on numerical arrays

The NumPy functions:

- np.sum()
- np.max(), np.min()
- np.mean(), np.median()
- np.diff() # takes the difference between elements
- np.cumsum() # cumulative sum

There are also "broadcast" functions that operate on all elements in an array

- my_array = np.array([12, 4, 6, 3, 4, 3, 7, 4])
- my_array * 2

- my_array2 = np.array([10, 9, 2, 8, 9, 3, 8, 5])
- my_array - my_array2

Boolean arrays

It is often to compare all values in an ndarray to a particular value

- my_array = np.array([12, 4, 6, 3, 4, 3, 7, 4])
- my_array < 5   # any guesses what this will return
    - array([False, True, False, True, True, True, False, True])

This can be useful for calculating proportions

- True == 1 and False == 0
- We can use the np.mean() function on a Boolean array to calculate a proportion
    - E.g.,   np.mean(position_array == "C")

Categorical Variable

PLAYER	POSITION	TEAM	SALARY
str	str	str	f64
"Paul Millsap"	"PF"	"Atlanta Hawks"	18.671659
"Al Horford"	"C"	"Atlanta Hawks"	12.0
"Tiago Splitter...	"C"	"Atlanta Hawks"	9.75625
"Jeff Teague"	"PG"	"Atlanta Hawks"	8.0
"Kyle Korver"	"SG"	"Atlanta Hawks"	5.746479

$$\text{Proportion centers} = \frac{\text{number of centers}}{\text{total number}}$$

Let’s review this in Jupyter!

Boolean masking

We can also use Boolean arrays to return values in another array

- This is called "Boolean masking" or "Boolean indexing"

```
my_array = np.array([12, 4, 6, 3, 1])
boolean_mask = np.array([False, True, False, True, True])

smaller_array = my_array[boolean_mask]
```

This can be useful for calculating statistics on data that meet particular criteria:

- `np.mean(my_array[my_array < 5])    # what does this do?`

Boolean masking

Suppose you wanted to get the average salary of NBA players who were centers

If you had these two ndarrays:

- Position: The position of all NBA players
- Salary: Their salaries

Could you do it?

Let's explore this in Jupyter!

Higher dimensional arrays

We can make higher dimensional arrays

- (matrices and tensors)

```
my_matrix = np.array([1, 2, 3], [4, 5, 6], [7, 8, 9])
my_matrix
```

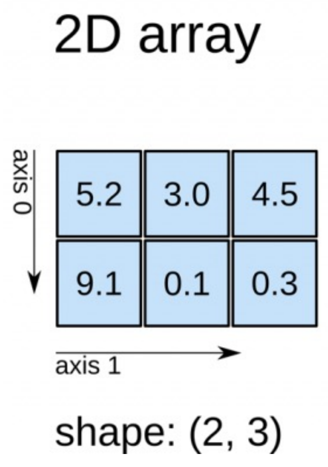
We can slice higher dimensional array

- my_matrix[0:2, 0:2]

We can apply operations to rows, columns, etc.

- np.sum(my_matrix, axis = 0)   # sum the values down rows

Let's explore this in Jupyter!



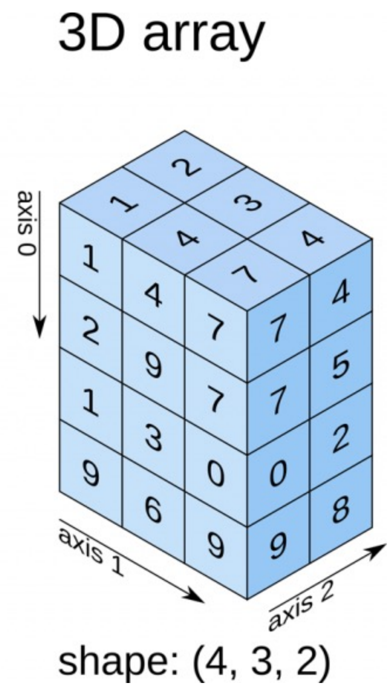
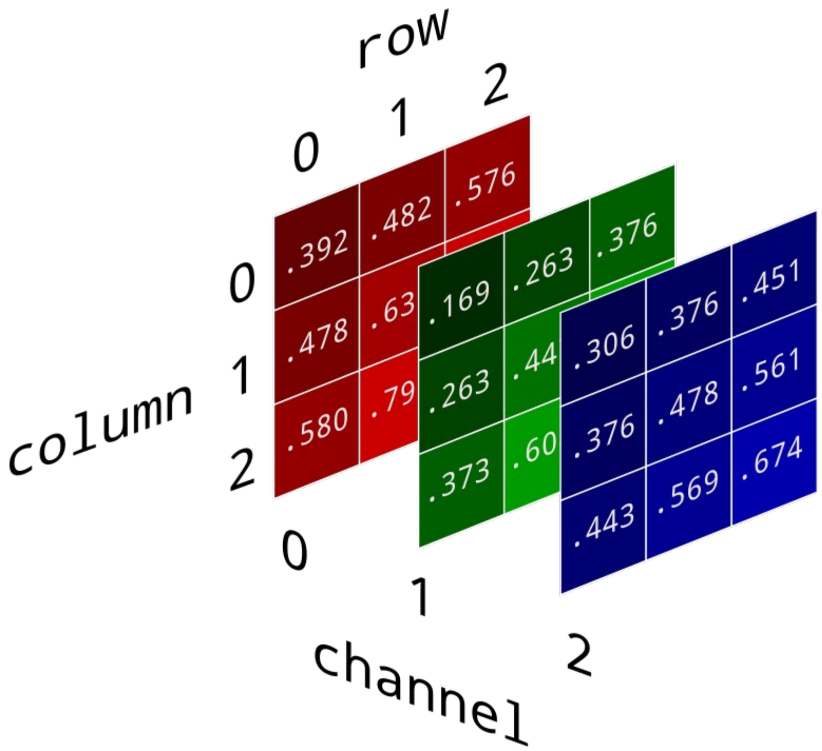


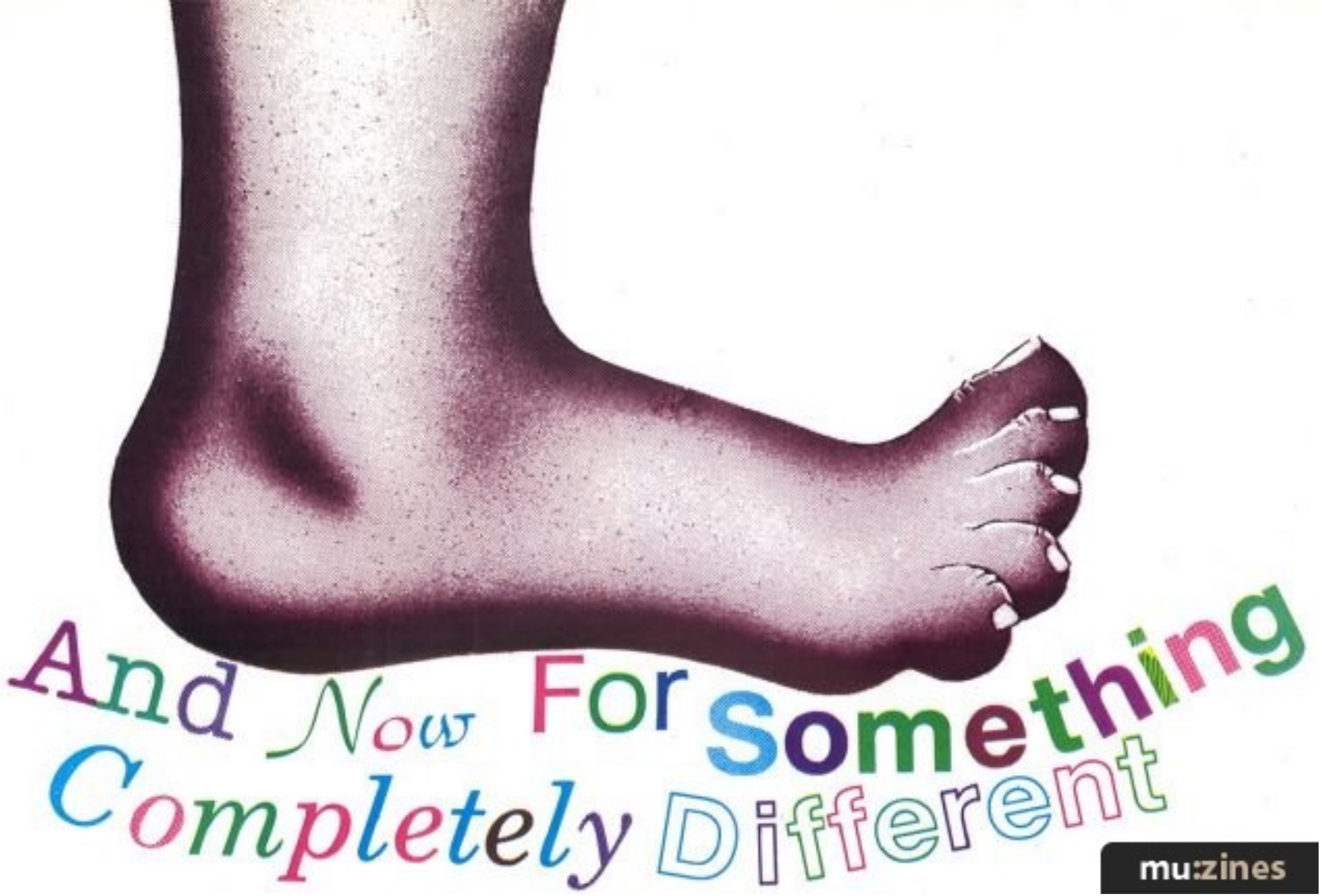
Image processing

3-dimemsional numerical arrays are often used to store digital images

We can use masking and other array operations to process images



Let's explore this in Jupyter!


And Now For Something Completely Different

Defining functions

Writing functions

We have already used many functions that are built into Python or are imported from different modules/packages.

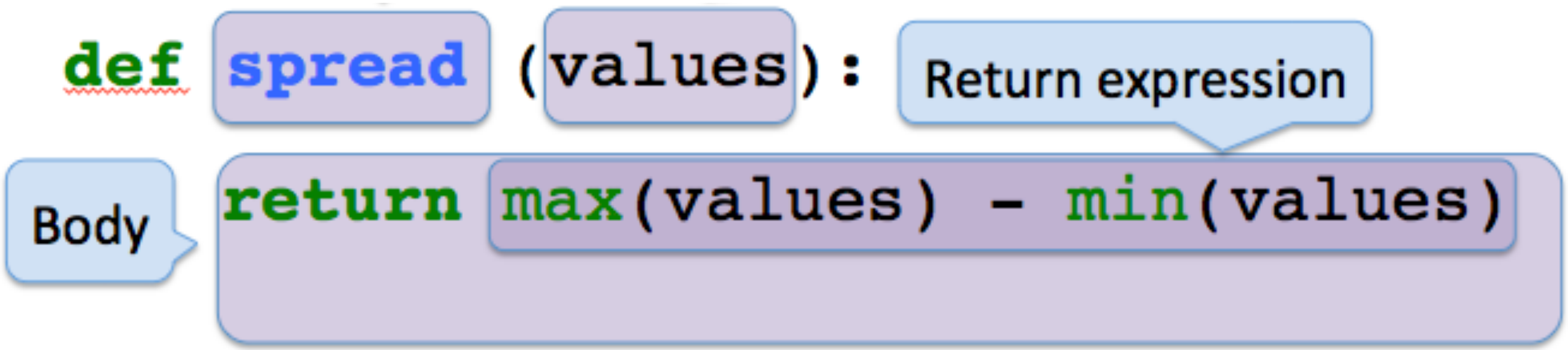
Examples…???

- sum()
- statistics.mean()
- np.diff()
- etc.

Let's now write our own functions!

Def statements

User-defined functions give names to blocks of code



Let's explore this in Jupyter!

Discussion questions

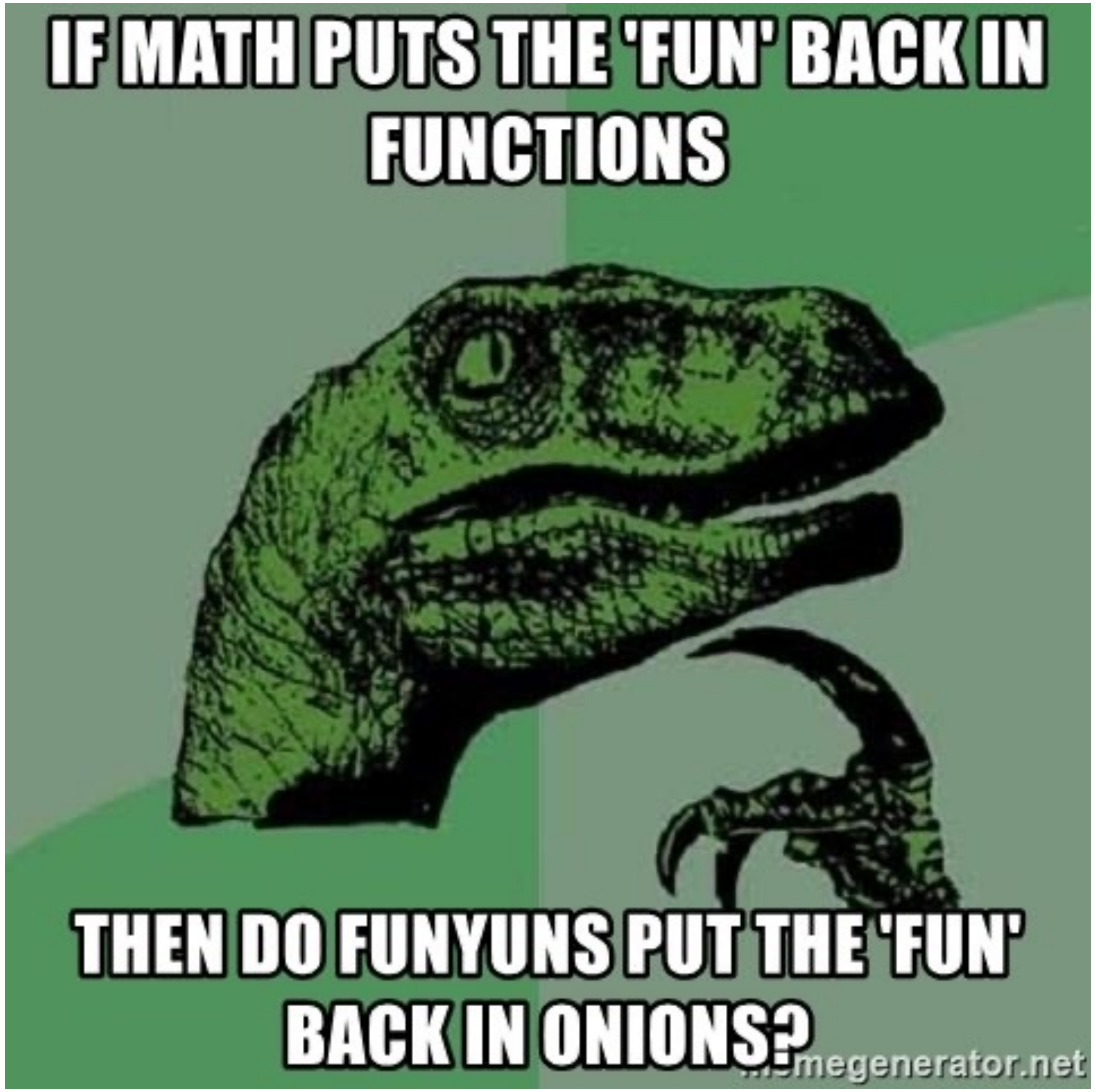
```
def f(s):
    return np.round(s/sum(s)*100, 2)
```

1. What does this function do?
2. What kind of input does it take?
3. What output will it give?
4. What's a reasonable name?

Let's explore this in Jupyter!

Next class…

Tables/DataFrames!!!


IF MATH PUTS THE 'FUN' BACK IN FUNCTIONS
THEN DO FUNYUNS PUT THE 'FUN' BACK IN ONIONS?
memegenerator.net