Queen Mary
University of London

QHP4701
Introduction to Data Science Programming

More on Numpy and Control Statements

Lecturer: Nikesh Bajaj, PhD
School of Physical and Chemical Sciences
http://nikeshbajaj.in

Lecture Outline

More on Numpy Arrays

- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Methods and Operations

- a = np.array([1,2,3, 4,5,6])

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Methods

- sum, max, min a.sum(), a.max(), a.min() or np.sum(a), np.max(a) …
- mean, var, std, a.mean(), a.var(), a.std()

- **Compute for b**

$$b = \begin{bmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \\ 6 \end{bmatrix}$$

NumPy : Methods and Operations

- a = np.array([[1,2,3],
  [4,5,6]])

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- a.sum() ➔ ?

- NumPy (by default) converts the array to 1D array (“flat-array”) and performs the operation

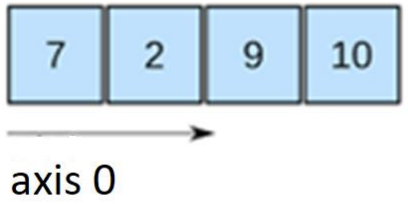
- *Axis: axis is used to represent each dimension of a NumPy array*

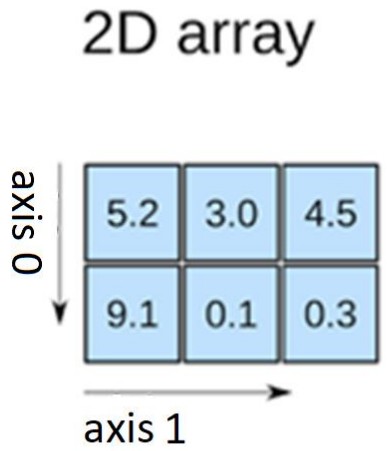
NumPy : Methods and Operations

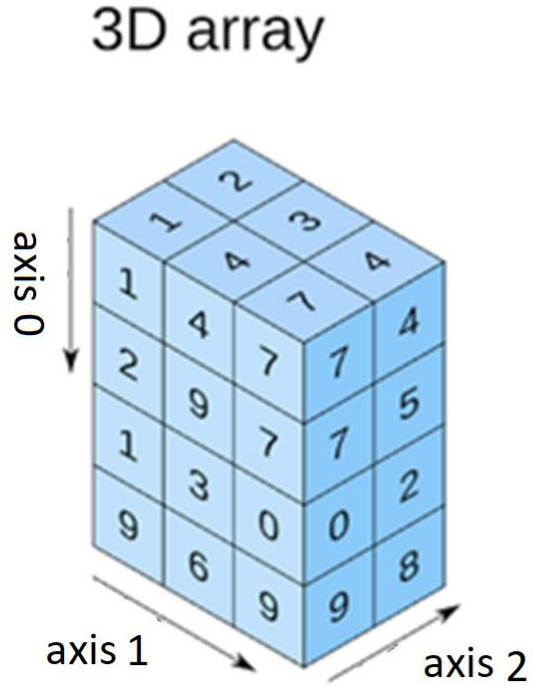
- a = np.array([[1,2,3],
  [4,5,6]])

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- a.sum(axis=?) ➔ ?







- *NumPy can carry out the operations along one of the axes*
- *Axis: axis is used to represent each dimension of a NumPy array*

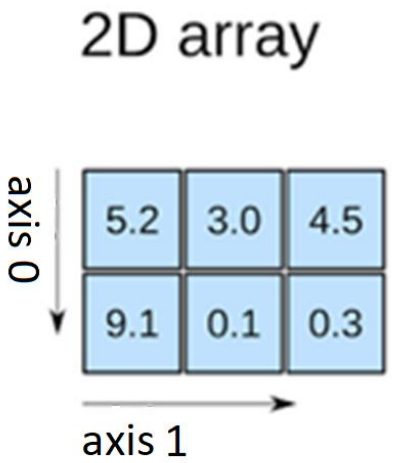
NumPy : Methods and Operations

- a = np.array([[1,2,3],
  [4,5,6]])

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Compute mean of each column by hand

a.mean(axis=?)



NumPy : Methods and Operations

- np.add
- np.subtract
- np.multiply
- np.divid
- np.power, np.sqrt, np.square
- np.log, np.log2, np.log10
- np.exp
- np.abs, np.absolute
- np.sin, np.cos, np.tan, np.arcsin, np.arccos
- .
- ….
- *Ref: https://numpy.org/doc/stable/reference/ufuncs.html#available-ufuncs*

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \\ 6 \end{bmatrix}$$

Lecture Outline

More on Numpy Arrays

- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Comparison

- a = np.array([[1,2,3],
  [4,5,6]])

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Comparison (==, =>, =<)

- a ==3, a>3
- a==b
- a==c

$$b = \begin{bmatrix} 1 & 1 & 3 \\ 3 & 5 & 5 \end{bmatrix}$$

$$c = \begin{bmatrix} 1 & 1 \\ 3 & 5 \end{bmatrix}$$

Lecture Outline

More on Numpy Arrays

- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Reshaping

- a = np.array([1,2,3, 4,5,6])

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

Changing the shape of a matrix

Re-organising the elements of an array

- Reshape a.reshape(2,3), a.reshape(2,-1), a.reshape(-1,2)
- Transpose a.T, a.transpose
- Flatten a.flatten a.ravel

Find out the difference

NumPy : Reshaping

- a = np.arange(20)

$$a = \begin{bmatrix} 0 \\ 1 \\ 2 \\ . \\ . \\ . \\ . \\ 17 \\ 18 \\ 19 \end{bmatrix}$$

- a.reshape(4, 5)
- a.reshape(4, -1)
- a.reshape(4, 3)
- a.reshape(2, 2, -1)

Lecture Outline

More on Numpy Arrays

- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Element-wise Operation(s)

- a = np.array([1,2,3, 4,5,6])
- b = np.array([2,2,2, 2,2,2])
- c = np.array([2,2,2,2,2]

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \qquad c = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

- a+b
- a+c?

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = ?$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = ?$$

NumPy : Element-wise Operation(s)

- a = np.array([1,2,3, 4,5,6])
- b = np.array([2,2,2, 2,2,2])

$$\begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} + \begin{bmatrix}2\\2\\2\\2\\2\\2\end{bmatrix} = ?$$

$$\begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} - \begin{bmatrix}2\\2\\2\\2\\2\\2\end{bmatrix} = ?$$

- a+b
- a-b
- a*b
- a/b
- a*a = a^2

$$\begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} * \begin{bmatrix}2\\2\\2\\2\\2\\2\end{bmatrix} = ?$$

$$\begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} * \begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} = ?$$

$$\begin{bmatrix}1\\2\\3\\4\\5\\6\end{bmatrix} / \begin{bmatrix}2\\2\\2\\2\\2\\2\end{bmatrix} = ?$$

NumPy : Broadcasting

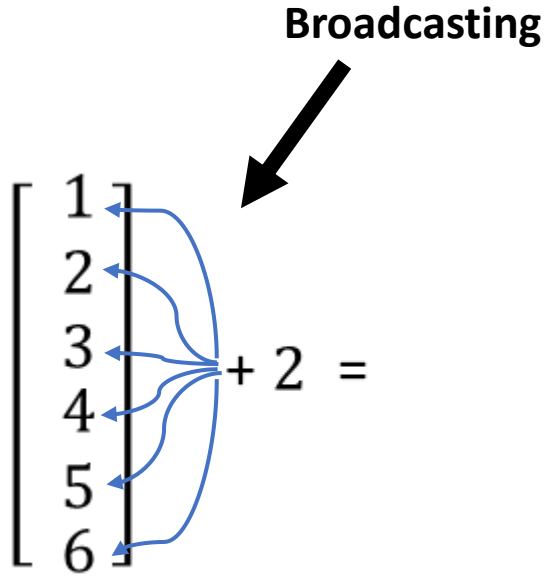
- a = np.array([1,2,3, 4,5,6])
- b = np.array([2,2,2, 2,2,2])

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

- a+b
- a+2

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + 2 = ?$$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} = ?$$



Lecture Outline

More on Numpy Arrays

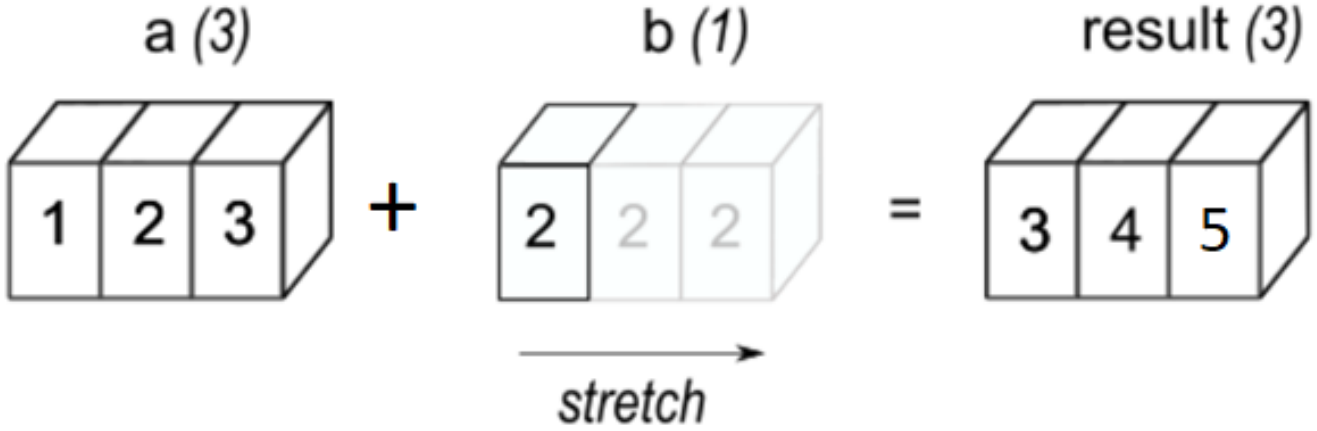
- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Broadcasting

- a = np.array([1,2,3, 4,5,6])
- b = np.array([2,2,2, 2,2,2])

$$a = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \qquad b = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

- a+b
- a+2



- *Broadcasting is a way to distribute the operation across axes.*
- *Python uses to avoid loops to* ***be more efficient***

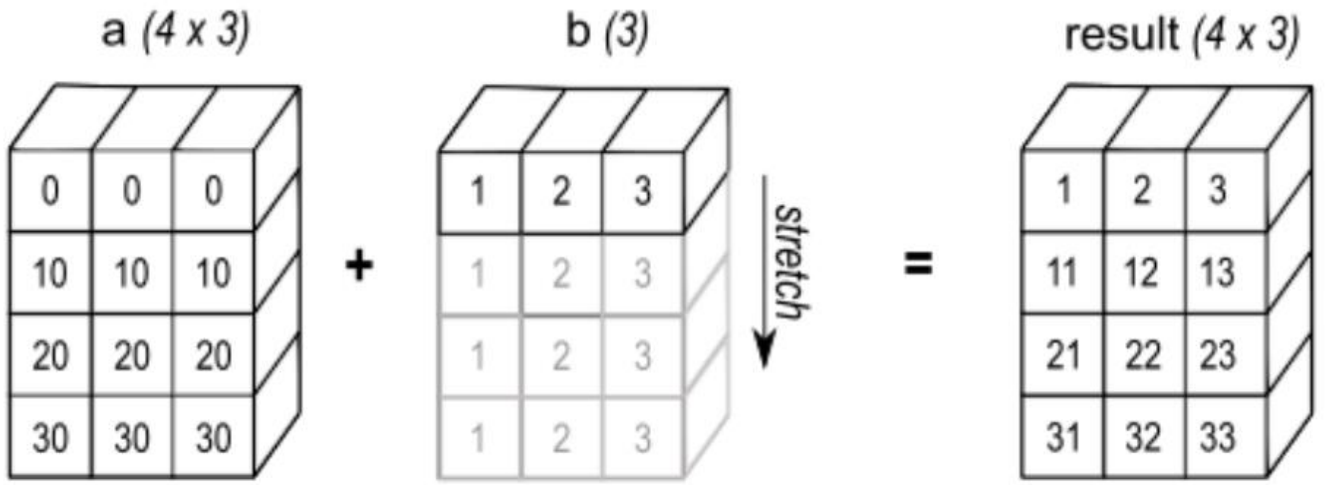
NumPy : Broadcasting

- a = np.array([[0,0,0], [10,10,10],[20,20,20],[30,30,30]])
- b = np.array([1,2,3])

$$a = \begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix}$$

$$b = [1 \;\; 2 \;\; 3]$$

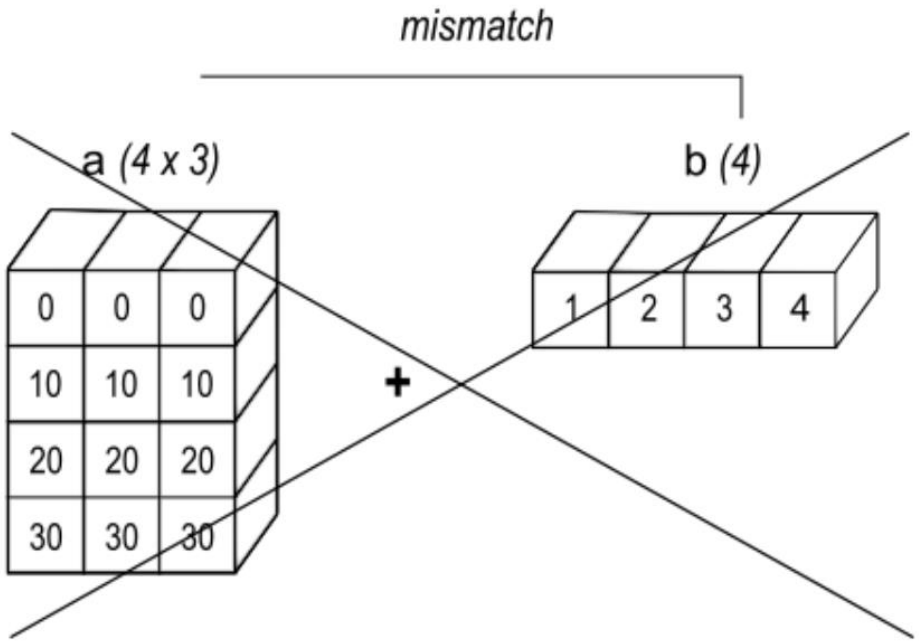
- a + b
-



NumPy : Broadcasting

- a = np.array([[0,0,0], [10,10,10],[20,20,20],[30,30,30]])
- b = np.array([1,2,3, 4])

- a + b
-



How can we add them?

NumPy : Broadcasting

- a = np.array([[0], [10],[20],[30]])

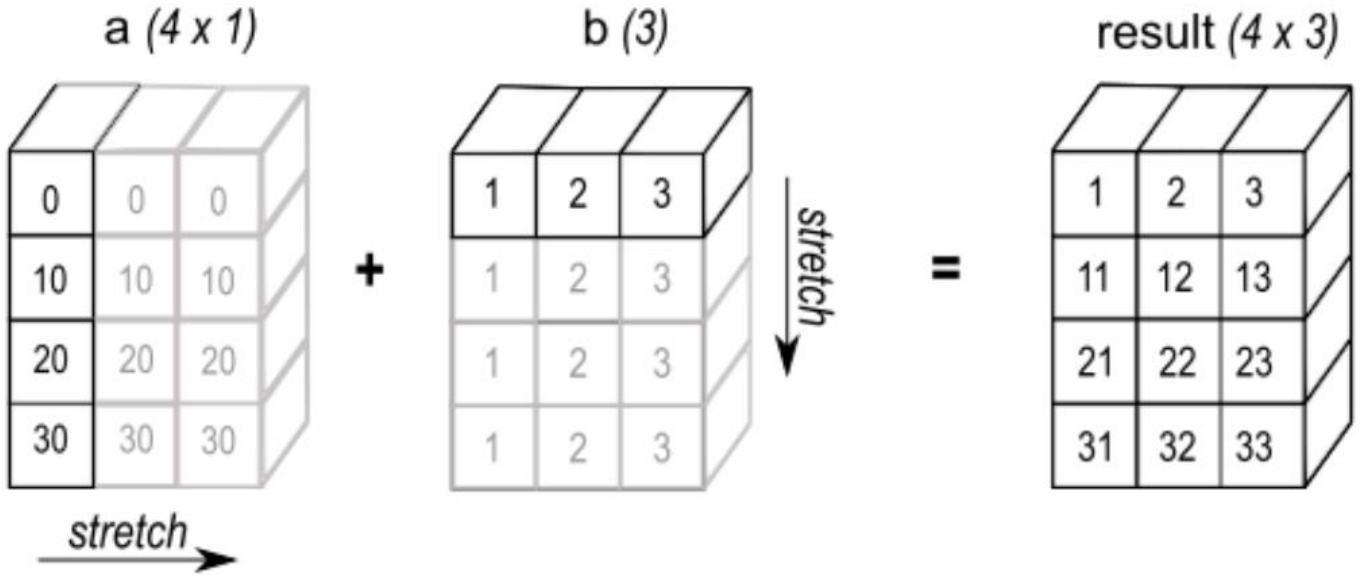
Or a = np.array([0,10,20,30]).reshape(-1,1)

$$a = \begin{bmatrix} 0 \\ 10 \\ 20 \\ 30 \end{bmatrix}$$

- b = np.array([1,2,3])

$$b = [1 \;\; 2 \;\; 3]$$

- a + b
-



NumPy : Broadcasting

- a = np.array([[0], [10],[20],[30]])

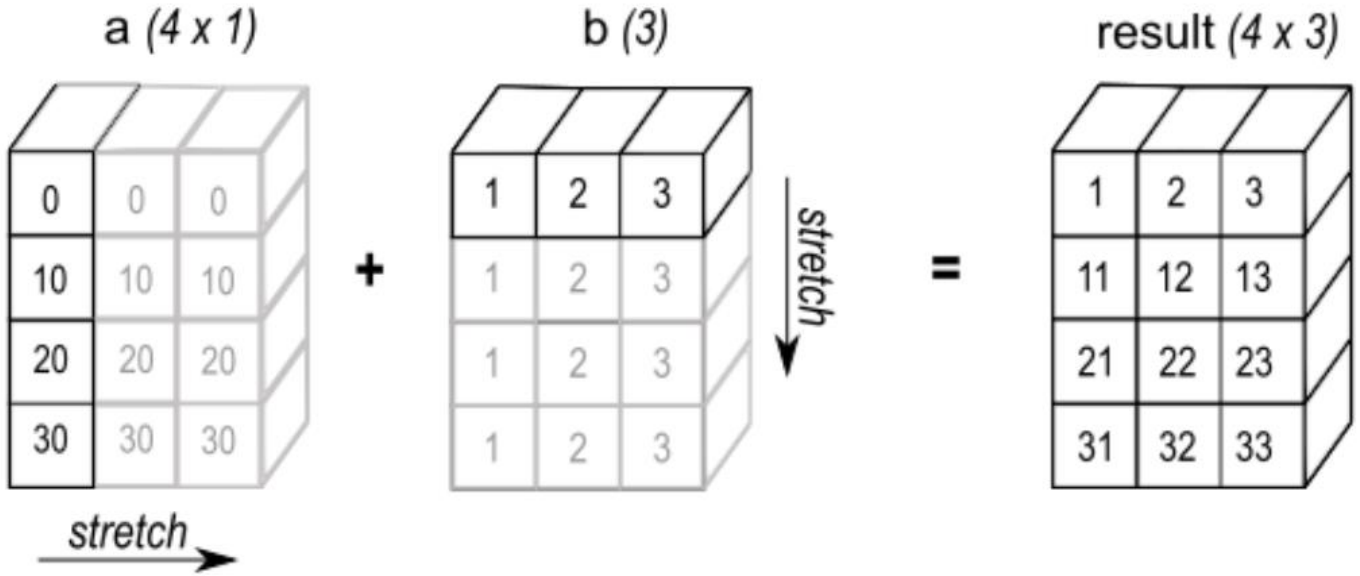
Or a = np.array([0,10,20,30]).reshape(-1,1)

$$a = \begin{bmatrix} 0 \\ 10 \\ 20 \\ 30 \end{bmatrix}$$

- b = np.array([1,2,3])

$$b = [1 \;\; 2 \;\; 3]$$

- a + b
-

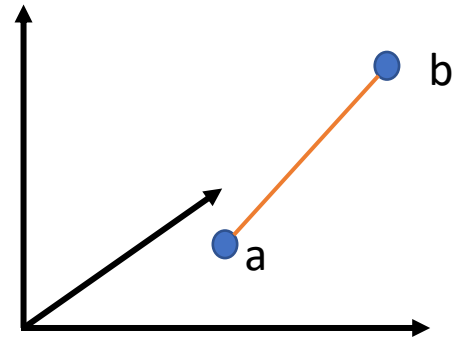


Lecture Outline

More on Numpy Arrays

- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Euclidean Distance



Compute Euclidean Distance between ***a*** and ***b***

- Dist: $\mathrm{D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$

$$\mathrm{a} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \qquad \mathrm{b} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

Compute distance between a and b

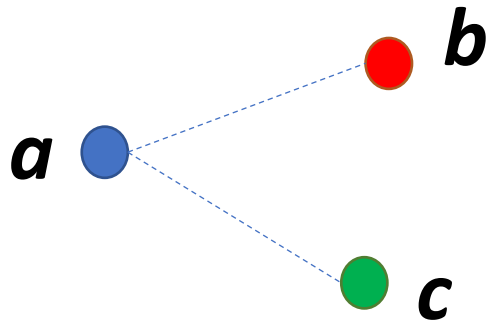
- ***a*** = np.array([1,2,3, 4,5,6])
- ***b*** = np.array([4,5,1, 2,3,6])

$$\mathrm{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \qquad \mathrm{b} = \begin{bmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \\ 6 \end{bmatrix}$$

NumPy : Euclidean Distance

Euclidean Distance between ***a*** and ***b***

- Dist: $\mathrm{D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$



Which is closer to a? b or c

is b closer to a than c?

- ***a*** = np.array([1,2,3, 4,5,6])
- ***b*** = np.array([4,5,1, 2,3,6])
- ***c*** = np.array([1,2,3, 2,3,6])

$$\mathrm{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} \qquad \mathrm{b} = \begin{bmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \\ 6 \end{bmatrix} \qquad \mathrm{c} = \begin{bmatrix} 4 \\ 4 \\ 3 \\ 3 \\ 3 \\ 6 \end{bmatrix}$$

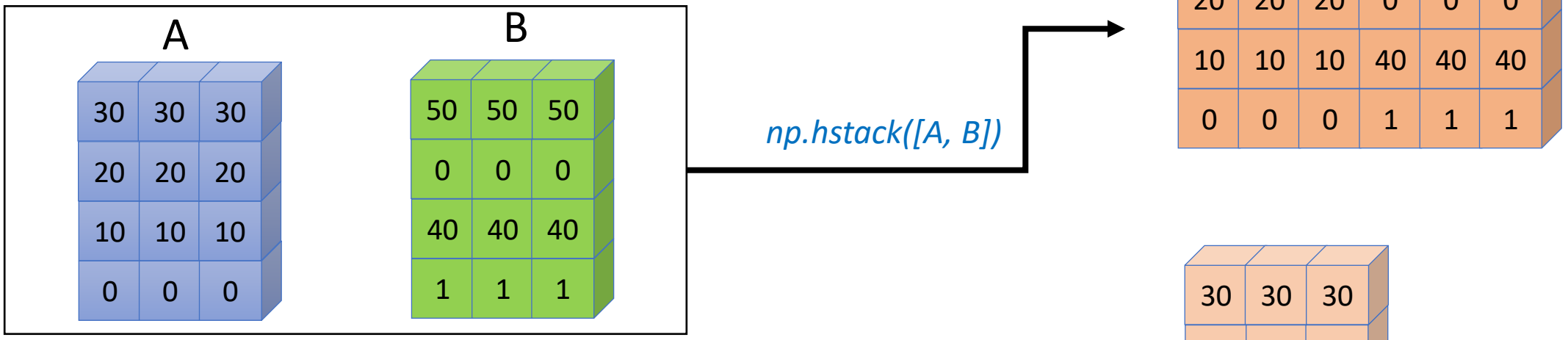
Lecture Outline

More on Numpy Arrays

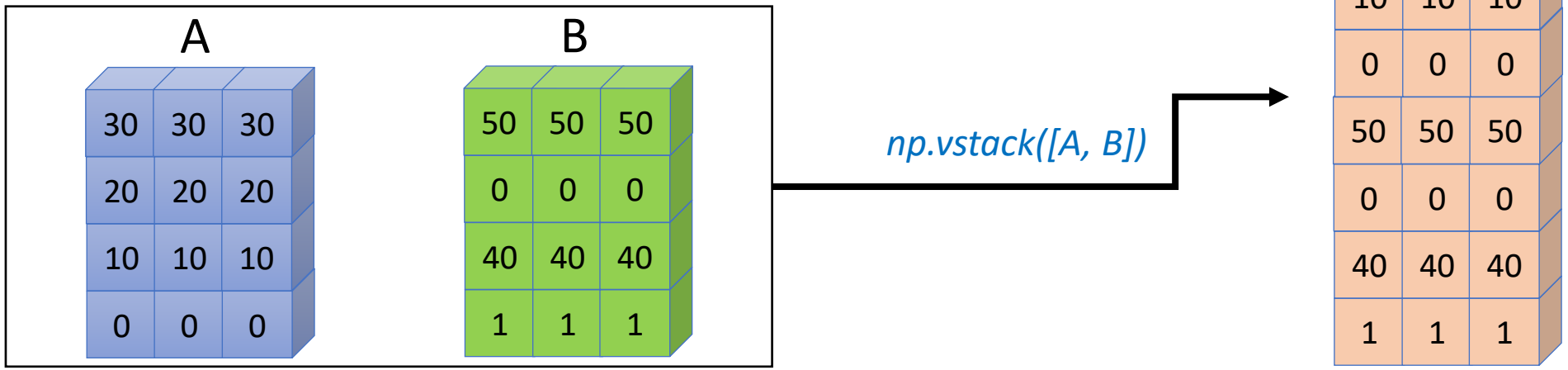
- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Stacking

- Horizontal Stacking: number of rows should match



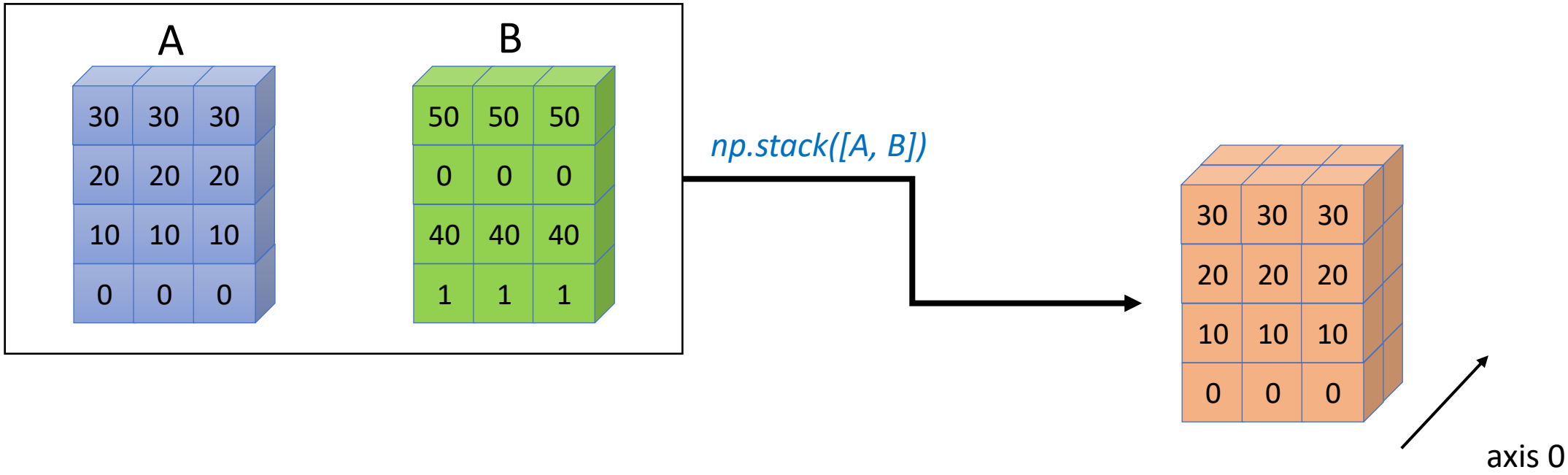
- Vertical Stacking : number of columns should match



NumPy : Stacking

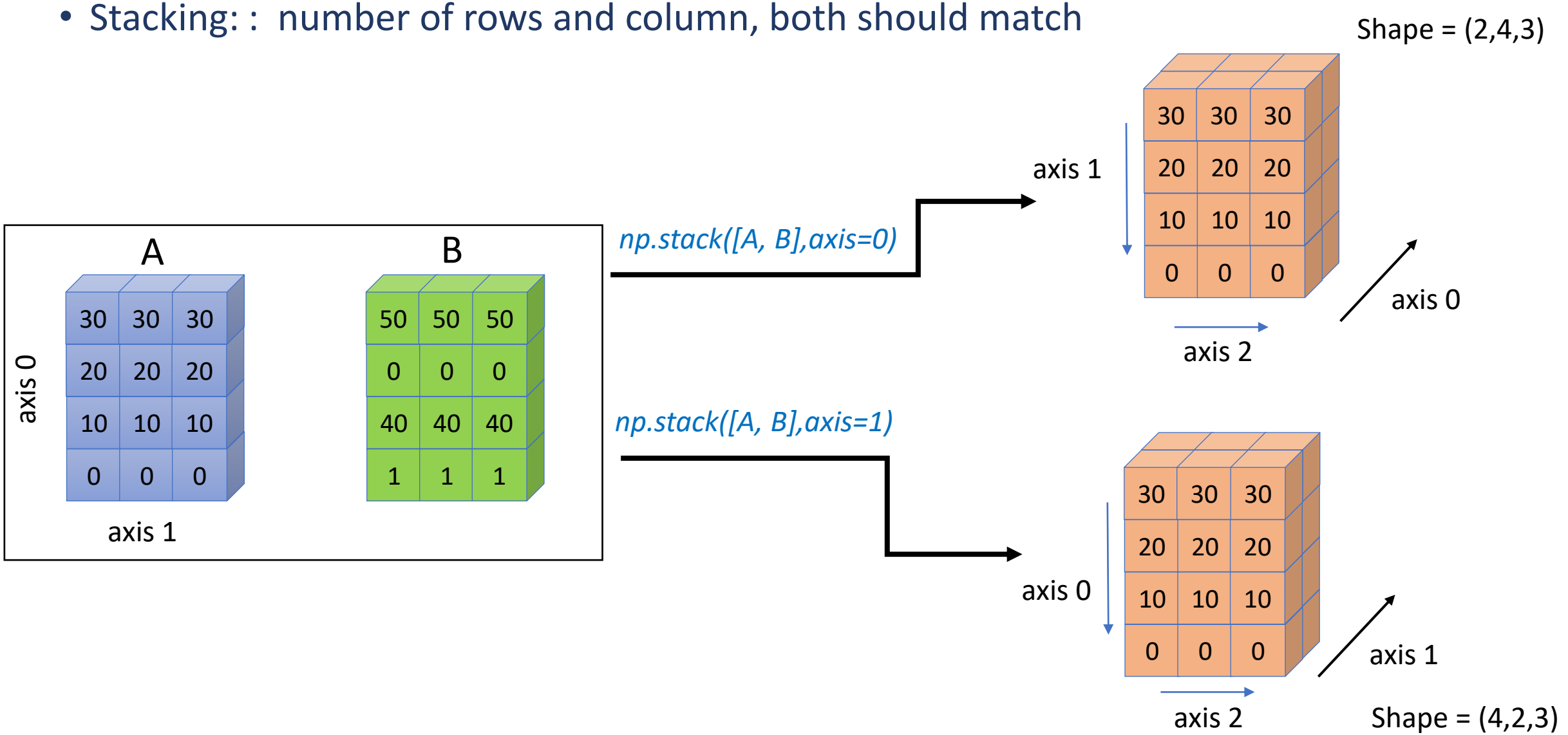
- Just Stacking: : number of rows and column, both should match

np.stack([A, B], axis=0)



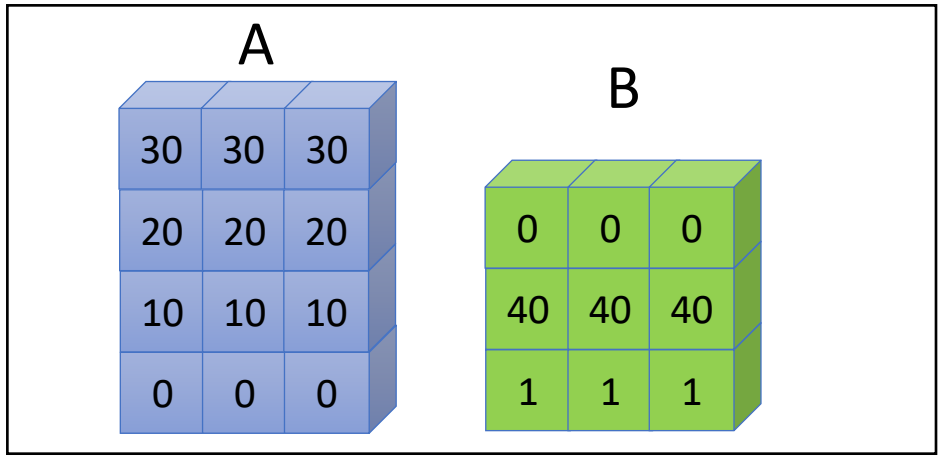
NumPy : Stacking

- Stacking: : number of rows and column, both should match

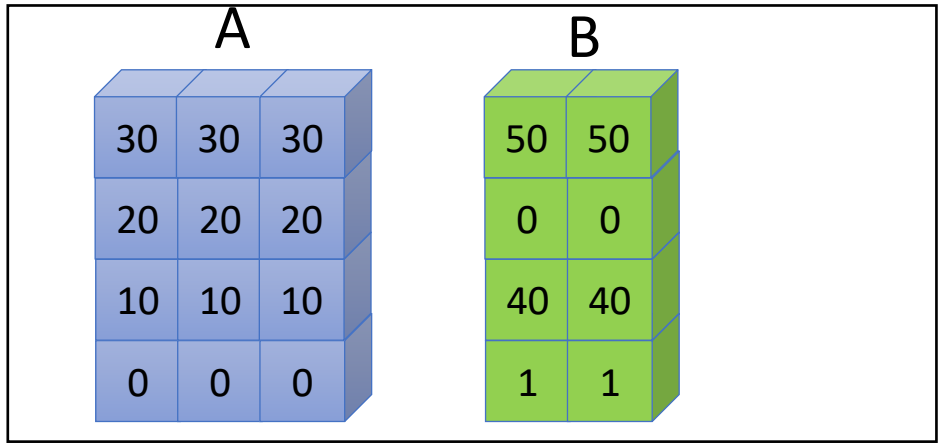


NumPy : Stacking-mismatch

- Horizontal Stacking



- Vertical Stacking



How can we add them?

Lecture Outline

More on Numpy Arrays

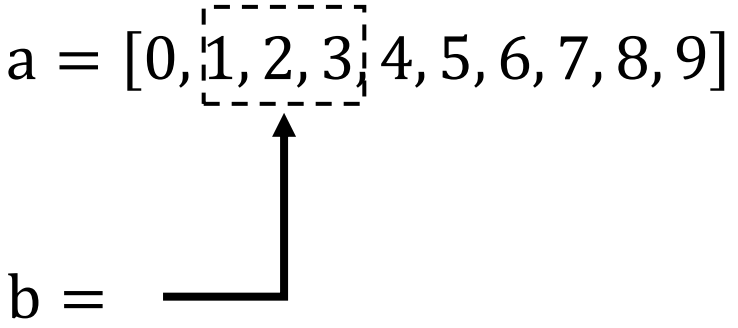
- Methods and Operations
- Comparison
- Reshaping
- Element-wise operation
- Broadcasting
- Euclidean Distance
- Stacking
- Copy

NumPy : Copy

- a = np.arange(10)
- b = a[1:4]
- a[1:4] = [0,0,0]

- print(a)
- print(b)

- c = a[1:4].copy()



c = [1,2,3]

- Next !!!
  - 3.1: More on NumPy
  - 3.2: Lab with Numpy Array

Queen Mary

University of London