

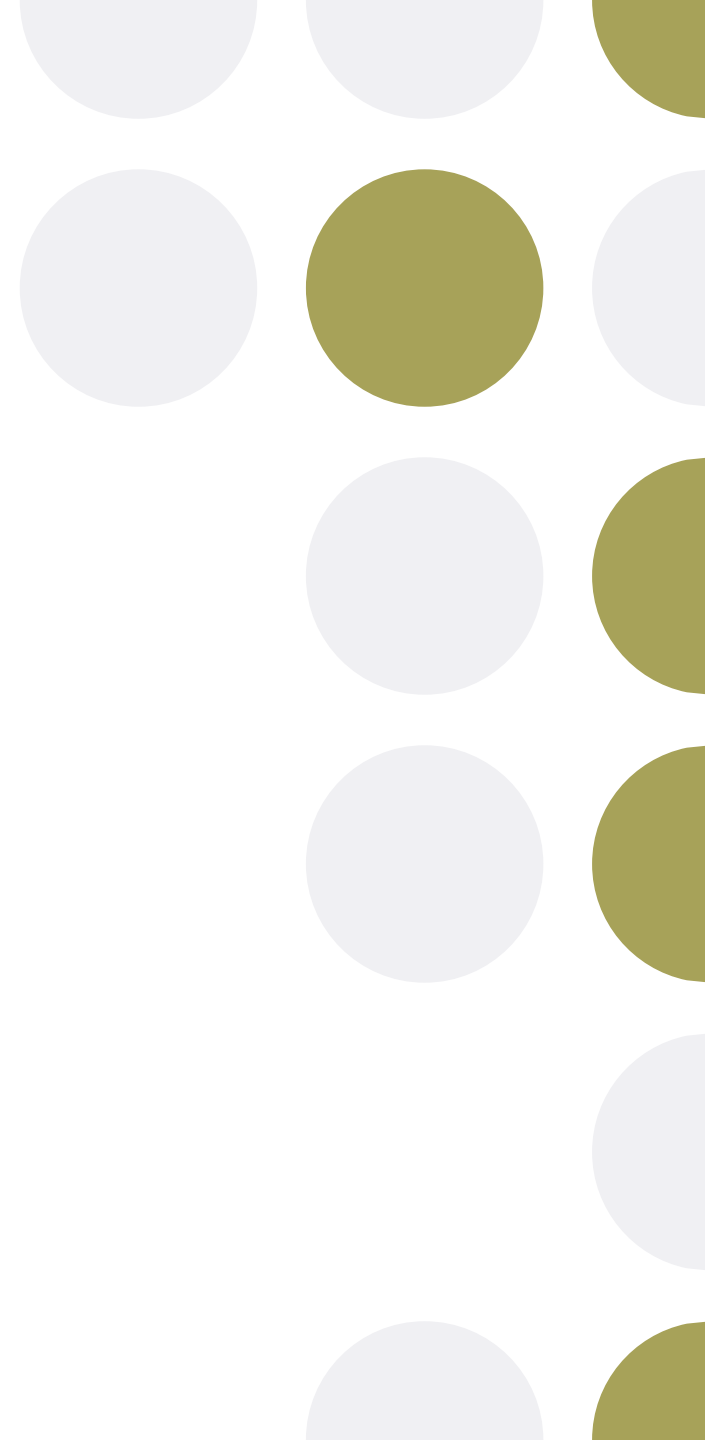


APAI 2024

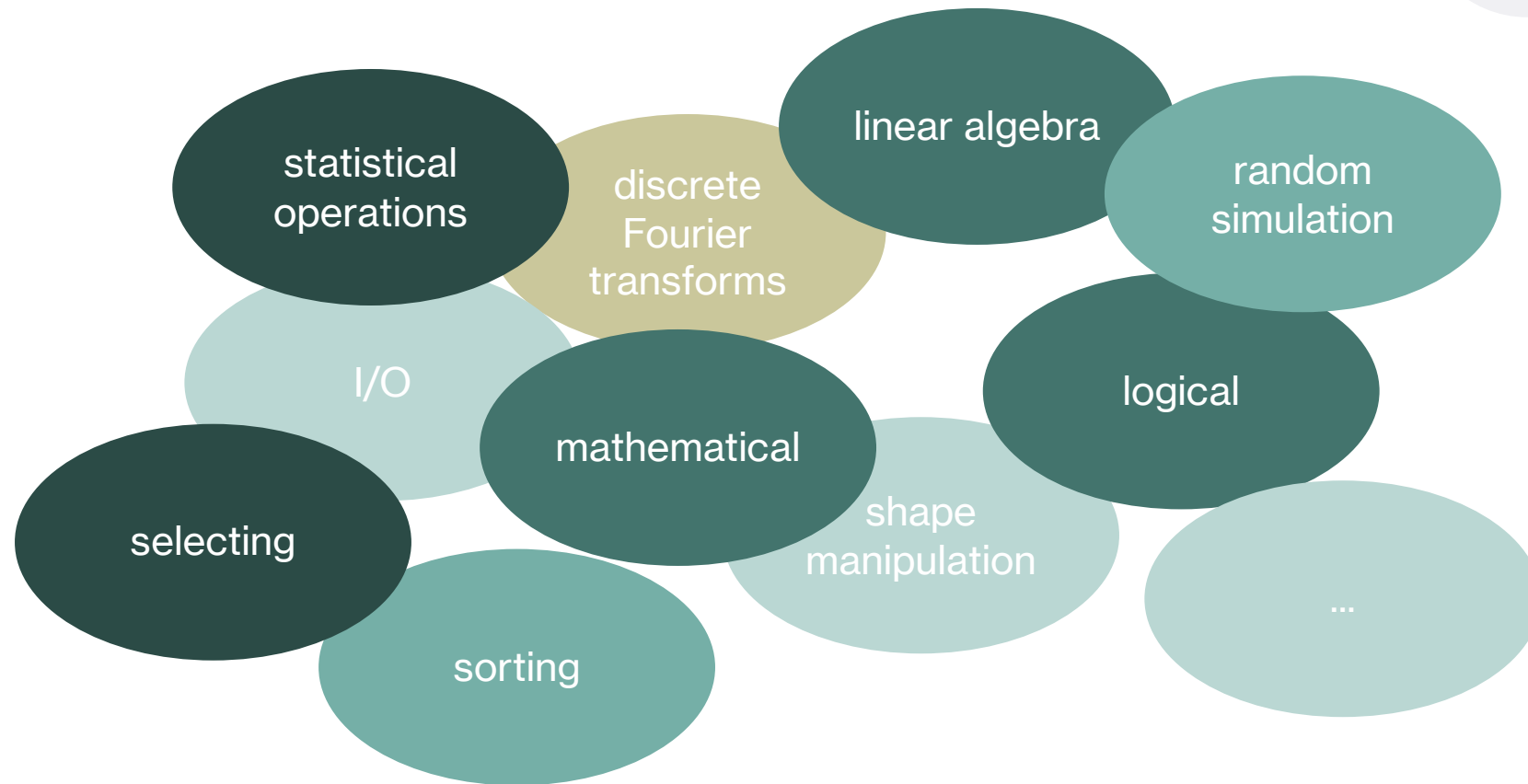
Lesson 3

Numpy

- The fundamental package for scientific computing with [Python](#)
 - It provides:
 - multidimensional array object
 - routines for fast operations on arrays
-



Numpy



Numpy

Install:

```
conda install numpy
```

or

```
pip install numpy
```

Numpy

Import:

```
import numpy as np
```



Numpy - Quickly Example

- Creating arrays filled with zeros, ones, or empty.

```
array1 = np.array([1, 2, 3])
print("Array1:", array1)

array2 = np.zeros((2, 3))
print("Array2 (filled with zeros):")
print(array2)

array3 = np.ones((3, 2))
print("Array3 (filled with ones):")
print(array3)
```

Numpy - Quickly Example

- Manipulating arrays: transposing

```
array5 = np.array([[1, 2], [3, 4]])  
print("Original array:")  
print(array5)  
  
print("Transposed array:")  
print(np.transpose(array5))
```

Numpy - Quickly Example

- Manipulating arrays: performing mathematical operations sum, mean, and element-wise square root.

```
print("Sum of array6:", np.sum(array6))  
print("Mean of array6:", np.mean(array6))  
print("Element-wise square root of array6:", np.sqrt(array6))
```


Numpy - Quickly Example

- Indexing and selection of elements.

```
array7 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  
print("Original array:")  
print(array7)  
  
print("Selecting second row:", array7[1])  
print("Selecting element at position (1, 2):", array7[1, 2])  
print("Selecting elements from the first two rows:", array7[:2])
```

Numpy - Quickly Example

- Linear algebra operations like matrix multiplication.

```
matrix1 = np.array([[1, 2], [3, 4]])  
matrix2 = np.array([[5, 6], [7, 8]])  
  
print("Matrix1:")  
print(matrix1)  
print("Matrix2:")  
print(matrix2)  
  
print("Matrix multiplication:")  
print(np.dot(matrix1, matrix2))
```

Numpy - Quickly Example

- Transforms (DFT) .

```
array8 = np.array([1, 2, 3, 4])  
print("Original array:")  
print(array8)  
  
print("Discrete Fourier Transform:")  
print(np.fft.fft(array8))
```

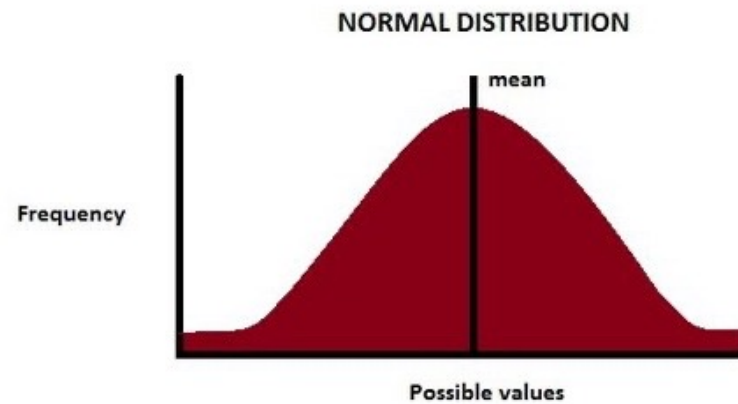
Numpy - Quickly Example

- Random number generation (uniform and normal distribution)

```
print("\nRandom number generation:")
print("Random numbers from a uniform distribution:")
print(np.random.rand(3, 3))

print("Random numbers from a normal distribution:")
print(np.random.randn(2, 2))
```

Numpy - Quickly Example



Numpy – Coding Now

- All the example here is implemented in the function called `task000()`
 - Go here https://github.com/clipsound/APAI2024_student
 - Clone or Download the repository
 - Create your virtual environment with all the specification on previous lesson (in the classroom we support pycharm and pip only).
-

Numpy – Coding Now

- Open file readme.MD and read the new task to implement
 - Use the file L0003_original.py to make your own code locally
-