Module - 2

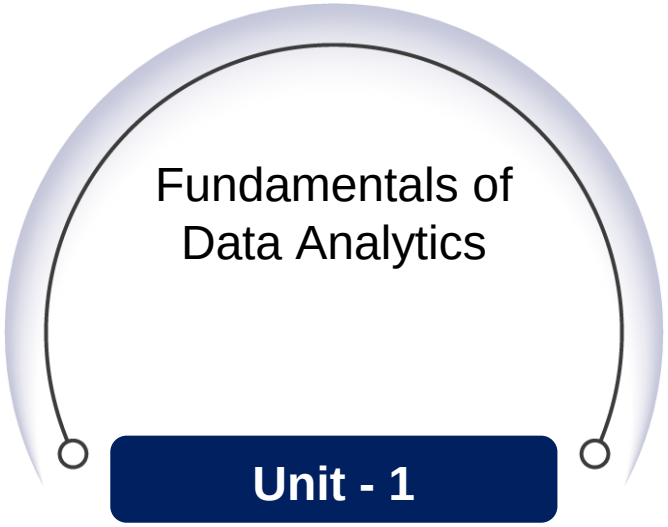
Data Analytics

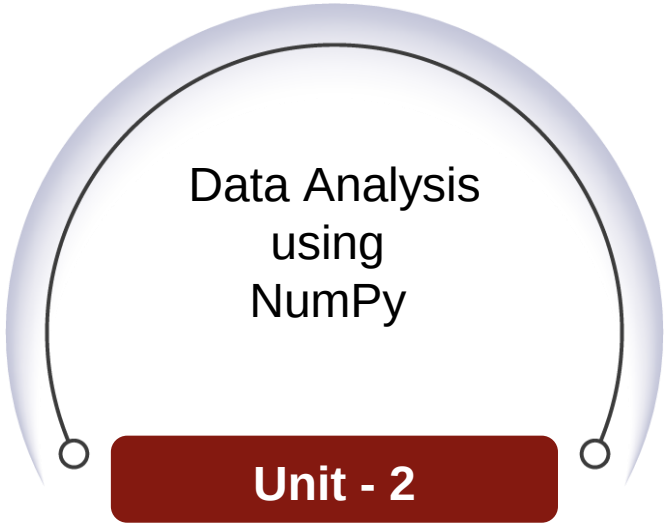
SAP

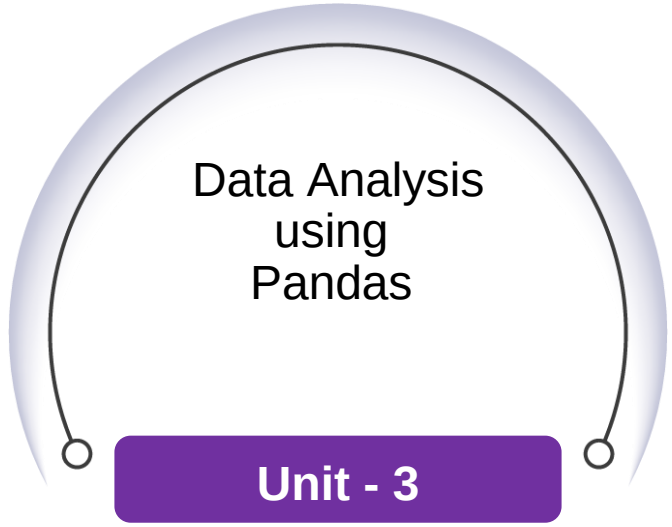
edunet foundation

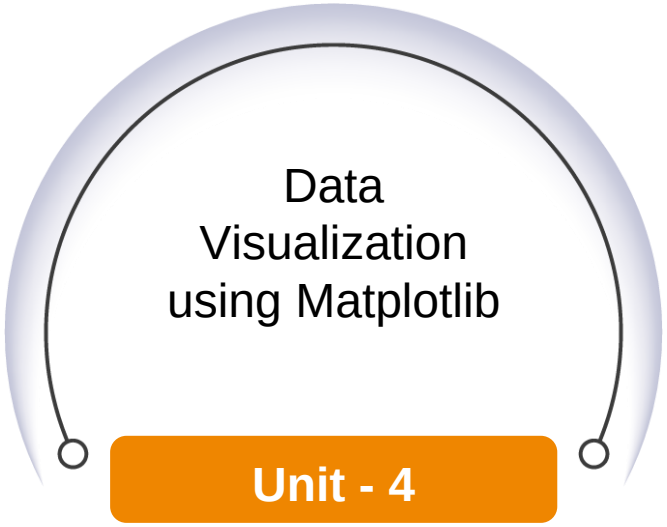
code unnati

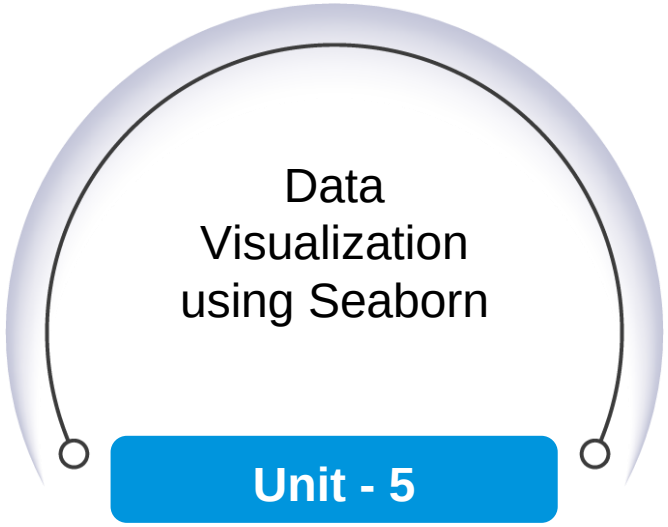
Units for Discussion











Unit - 2

Data Analysis using NumPy

DISCLAIMER

The content is curated from online/offline resources and used for educational purpose only.

How we can Calculate Mean of a given Data?

data = [12, 18, 25, 30, 15, 21]

```
data = [12, 18, 25, 30, 15, 21]

# Calculate the mean
mean = sum(data) / len(data)

print("Mean:", mean)
```

```
Mean: 20.166666666666668
```

Formula -

$$\mu=(nx1+x2+x3+\ldots+xn) / n$$

Mean = Sum of all data points / Number of data points

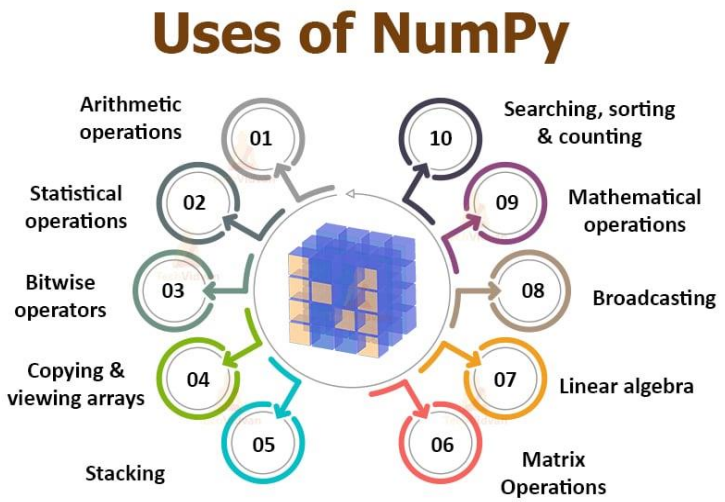
Lets simplify this →

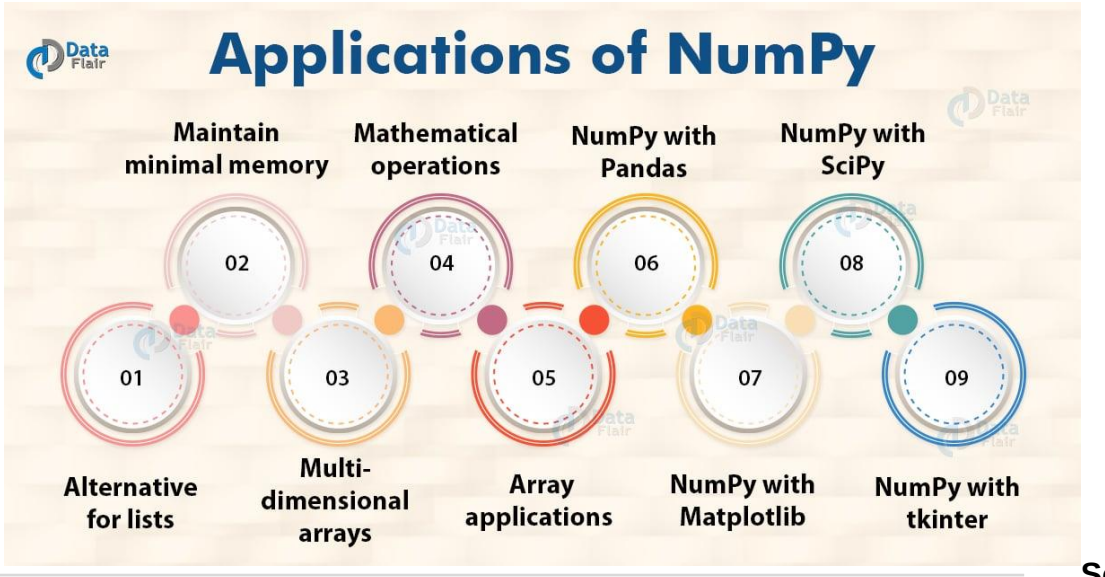
You can easily calculate the mean using the NumPy library in Python.

```
:  import numpy as np

   data = np.array([12, 18, 25, 30, 15, 21])

   # Calculate the mean using NumPy
   mean = np.mean(data)

   print("Mean:", mean)
```

```
Mean: 20.166666666666668
```

The **np.mean()** function calculates the Mean of an array.





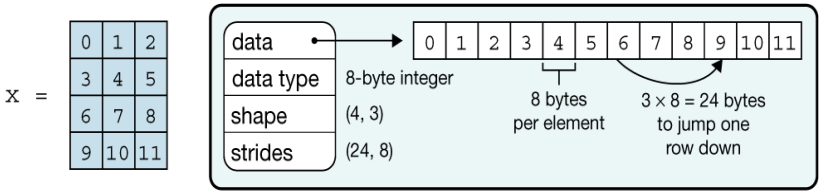
Source :

Reference

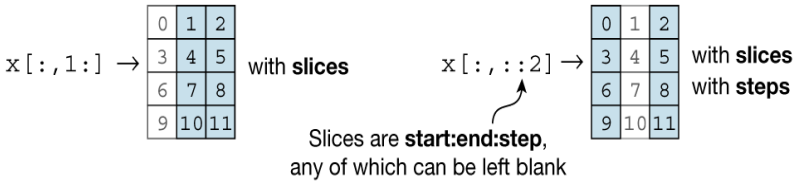
Reference

edunet foundation

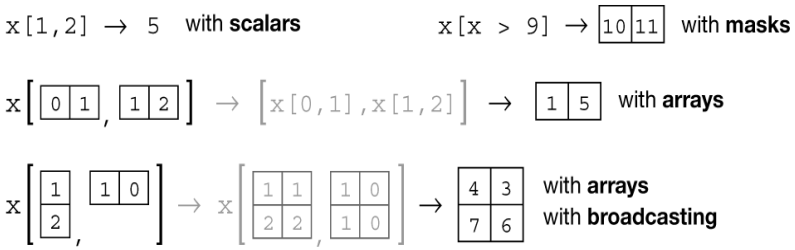
a Data structure



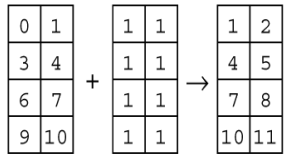
b Indexing (view)



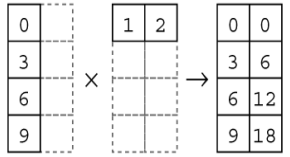
c Indexing (copy)



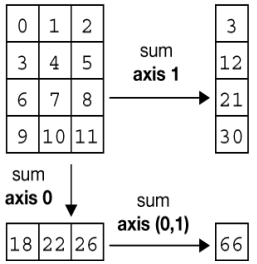
d Vectorization



e Broadcasting



f Reduction



g Example

```
In [1]: import numpy as np

In [2]: x = np.arange(12)

In [3]: x = x.reshape(4, 3)

In [4]: x
Out[4]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11]])

In [5]: np.mean(x, axis=0)
Out[5]: array([4.5, 5.5, 6.5])

In [6]: x = x - np.mean(x, axis=0)

In [7]: x
Out[7]:
array([[-4.5, -4.5, -4.5],
       [-1.5, -1.5, -1.5],
       [ 1.5,  1.5,  1.5],
       [ 4.5,  4.5,  4.5]])
```

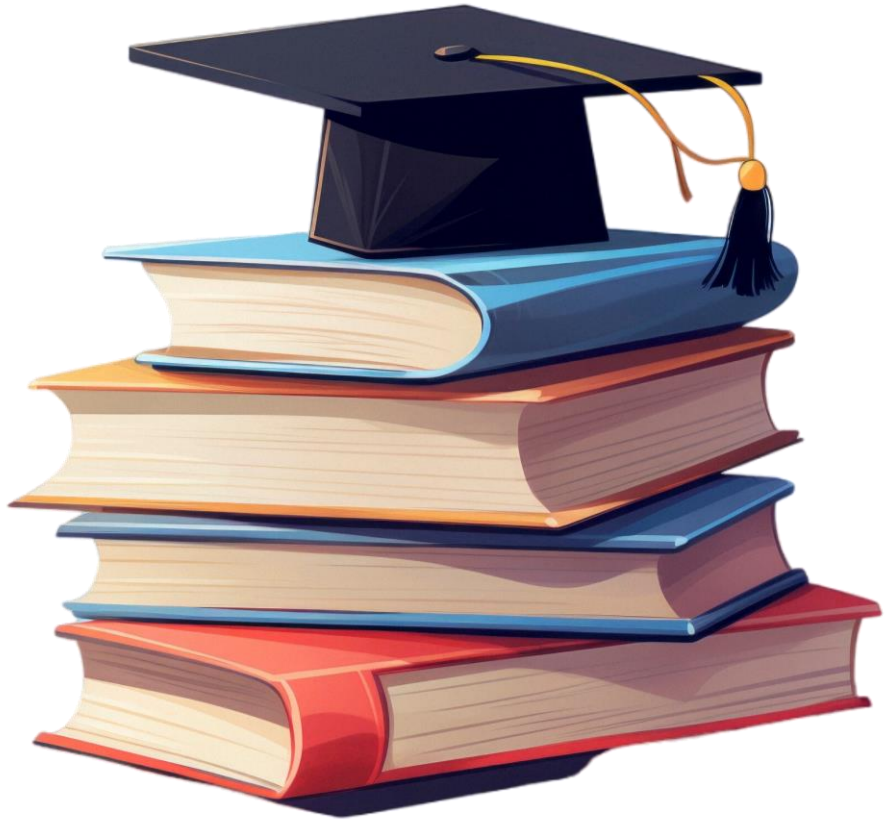
NumPy

Source :

[Reference]

Learning Objectives

- Introduction to NumPy
- Installing NumPy
- NumPy Datatypes
- NumPy Array
- Creating NumPy Array
- NumPy Array Attributes
- Indexing and Slicing
- NumPy Array Manipulation
- Broadcasting
- NumPy Statistical Functions
- I/O with NumPy



Source :

Reference

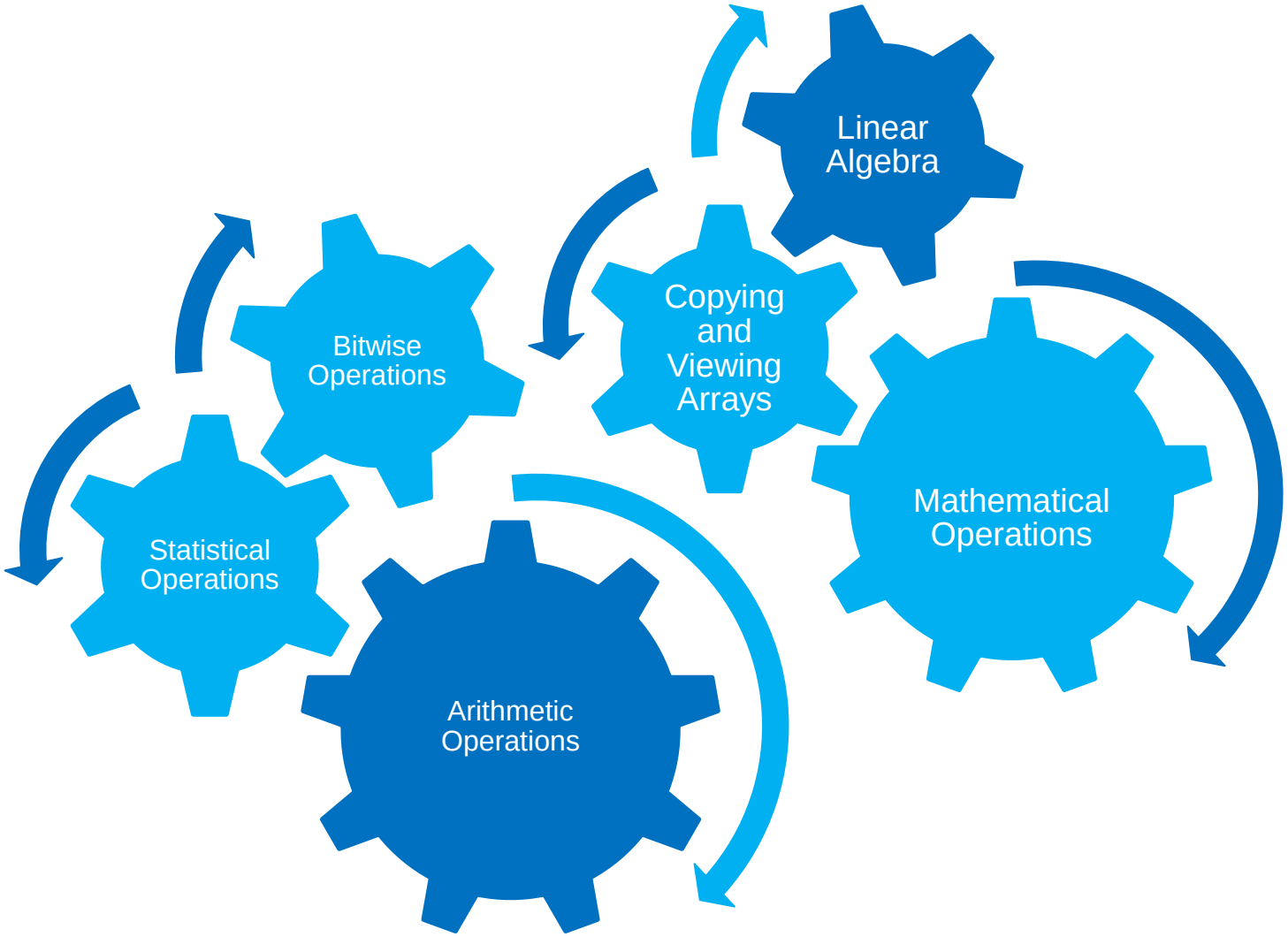
Introduction to NumPy

- NumPy Stands for Numerical Python
- It is an open-source Python library used for working with arrays.
- It has functioning for working with linear algebra, the Fourier transform, and matrices.
- NumPy is written in C, making NumPy arrays faster and more memory efficient than Python lists.
- NumPy was created in 2005 by Travis Oliphant.

Source :

Reference

Why NumPy?


Linear Algebra
Copying and Viewing Arrays
Bitwise Operations
Statistical Operations
Arithmetic Operations
Mathematical Operations

Installing NumPy

- You may use Command Prompt/Terminal
- You need pip/conda to install various libraries
  - pip install numpy
  - conda install numpy

Note: It is pre-installed if Anaconda Software is used

Why NumPy is Fast?

NumPy arrays are faster for a variety of reasons.

- Python is a dynamic language that is interpreted, converted to bytecode, and then executed by a Python interpreter.
- NumPy, on the other hand, is written in C, which results in a faster execution time.

NumPy Datatypes

- The data types are used to describe a variable with a specific type for identifying the variable and allowing the given types of data.
- The following list of numeric data types-

bool_

This is used to represents the Boolean value indicating true or false. It is stored as a byte.

int_

This is the default type of an integer. It is identical to long type in C that mainly contains 64 bit or 32-bit integer.

float_

It is identical to float64

NumPy Datatypes

complex 128

It is used to represent complex numbers in which the real and imaginary parts each have 64 bits.

float64

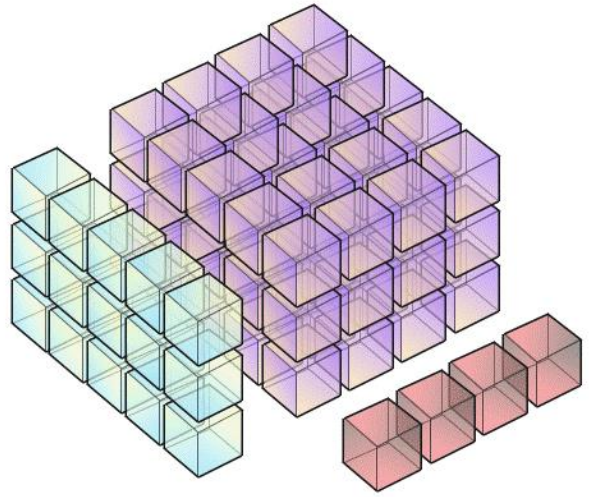
It is the double precision float. 11 bits are reserved for the exponent, 52 bits are reserved for mantissa, 1 bit is used for the sign.

int

It indicates the integers used for indexing.

NumPy Array

- Python lists and NumPy arrays are very different from one another, despite some similarities.
- A grid of identically typed values that are indexed by a pair of nonnegative integers is referred to as a NumPy array.
- The array object in NumPy is called ndarray.
- We can create a NumPy ndarray object by using two main method:
    1. From existing data (list, tuples) using np.array() method.
    2. Using built in methods like zeros(), linspace(), ones() etc.



NumPy arrays

Source :

Reference

Creating a NumPy Array using np.array() method

```
import numpy as np

arr = np.array(42)

print(arr)
```

```
42
```

0-D Arrays

- The elements of an array are 0-D arrays, also known as Scalars. Each array value is a 0-D array.

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)
```

```
[1 2 3 4 5]
```

1-D Arrays

- A uni-dimensional or 1-D array is one that has 0-D arrays as its elements.
- These are the most common and fundamental arrays.

Creating a NumPy Array using np.array() method

```
import numpy as np

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

print(arr)
```

```
[[1 2 3]
 [4 5 6]]
```

2-D Arrays

- A 2-D array is one that contains 1-D arrays as elements.
- These are frequently used to represent matrices or second order tensors.

```
import numpy as np

arr1 = np.array([[[2,17], [45, 78]], [[88, 92],
            [60, 76]],[[76,33],[20,18]]])
print("Create 3-d array:",arr1)
```

```
Create 3-d array: [[[ 2 17]
  [45 78]]

 [[88 92]
  [60 76]]

 [[76 33]
  [20 18]]]
```

3-D arrays

- A 3-D array is one that contains 2-D arrays (matrices) as its elements.
- These are frequently used to represent a tensor of third order.

NumPy Array Attributes

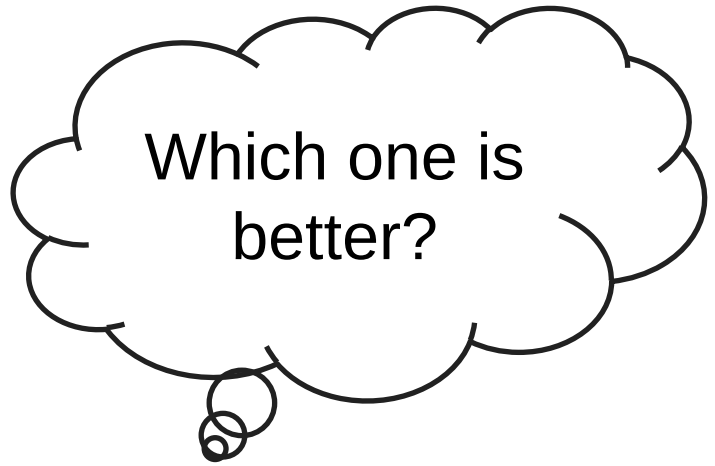
- In NumPy, attributes are properties of NumPy arrays that provide information about the array's shape, size, data type, dimension, and so on.
- Here are some of the commonly used NumPy attributes:

Attributes	Description
shape	It returns the size of the array in each dimension.
ndim	It returns number of dimension of the array
size	It returns number of elements in the array
dtype	It returns data type of elements in the array
itemsize	It returns the size (in bytes) of each elements in the array

NumPy Array Vs Python List

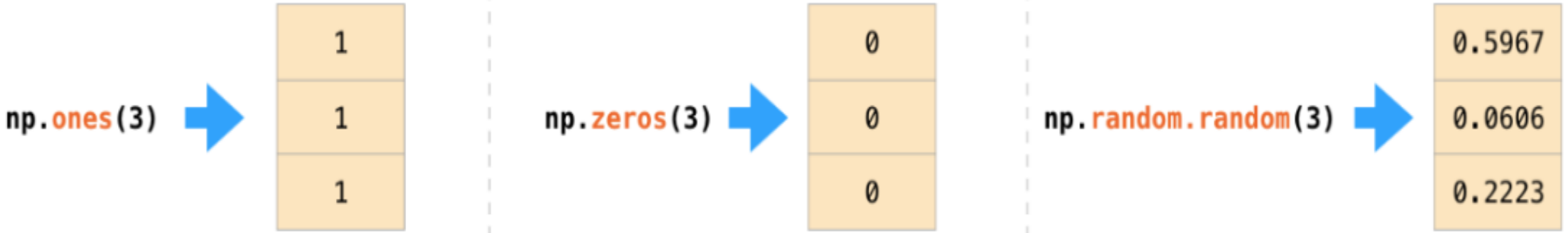
NumPy array is similar to a Python list. There are three main reasons why we would use NumPy array instead of lists in Python. These reasons are:

- Less memory usage: The Python NumPy array consumes less memory than lists.
- Less execution time: The NumPy array is pretty fast in terms of execution, as compared to lists in Python.
- Functionality : SciPy and NumPy have optimized functions such as linear algebra operations built in.



Different ways of creating NumPy Array

- NumPy has built-in functions for creating arrays from scratch. For example, zeros(shape) will create an array filled with 0 values with the specified shape.



Source :

Reference

Different ways of creating NumPy Array

- Here are some of the commonly used methods to initialize the values of array.

Method	Description
zeros()	zeros(shape) will create an array filled with 0 values with the specified shape.
ones()	ones(shape) will create an array filled with 1 values with the specified shape.
linspace()	linspace() method creates an array with evenly spaced elements over an interval.
arange()	arange() method creates an array with evenly spaced elements as per the interval.
full()	full() method creates a new array of given shape and type, filled with a given fill value.

Indexing

- In NumPy, each element in an array is associated with a number. The number is known as an array index.
- Index of a NumPy array starts with ‘0’, second element have 1 and so on.
- Indexing is used to access individual elements. With NumPy indexing, you can also extract entire rows, columns, or planes from multidimensional arrays.
- NumPy supports both positive and negative index. To access last item we have to use -1 index, -2 to the second last item and so on.

Element of array	2	3	11	9	6	10
Index	0	1	2	3	4	5
Negative Index	-6	-5	-4	-3	-2	-1

Indexing 1-D Array

- One-dimensional arrays can be indexed, iterated over, much like lists and other Python sequences.
- Suppose we have NumPy array named data:

1	2	3

- data[0] – to access first element i.e 1.
- data[1] – to access second element i.e 2.

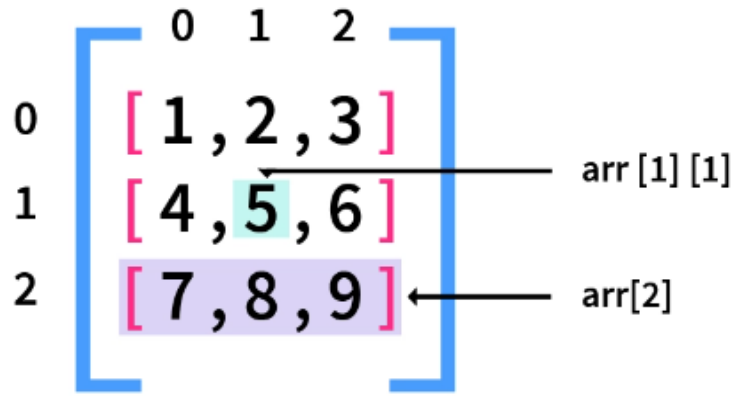
Source :

Reference

Indexing 2-D Array

- Each element in a 2-D array is represented by two indices, the row and the column. 2D arrays in python are zero-indexed, which means counting indices start from zero rather than one; thus, zero is the first index in an array in python.
- Syntax to access element is:

```
array_name[row_ind, col_ind]
```



- Where array_name is the array name, row_ind is row index and col_ind is the column index of the element

Slicing an Array

- Slicing a NumPy array means accessing the subset of the array. It means extracting elements from an array in the specific range.
- Slicing is performed using indexing. So that means slicing can be performed using positive indexing and negative indexing.

Slicing syntax is as follows:

Syntax: arr_name[start:stop:step]

Start: Starting index Stop: Ending index Step: Difference between the index

Slicing an Array

Slicing 1D NumPy Arrays

```
: import numpy as np
arr = np.arange(6)
print("array arr:",arr)
print("sliced element of array:",arr[1:5])
```

```
array arr: [0 1 2 3 4 5]
sliced element of array: [1 2 3 4]
```

Slicing a 2D Array

```
import numpy as np
arr=np.arange(12)
arr1=arr.reshape(3,4)
print("Array arr1:\n",arr1)
print("\n")
print("elements of 1st row and 1st column upto last column :\n",arr1[1:,1:4])
```

```
Array arr1:
 [[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]


elements of 1st row and 1st column upto last column :
 [[ 5  6  7]
 [ 9 10 11]]
```

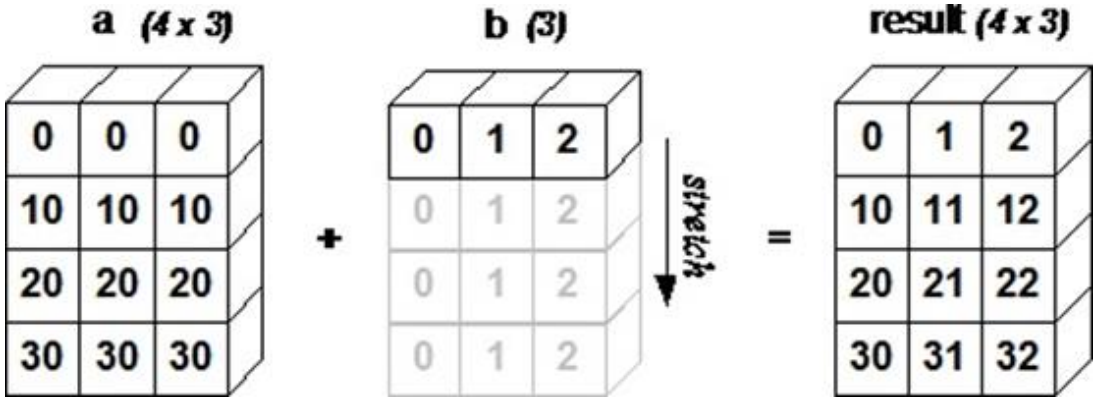
NumPy Array Manipulation

- The NumPy package is a multi-purpose prebuilt package developed majorly to process and aid in the data manipulation of arrays.
- NumPy array manipulation functions allow us to modify or rearrange NumPy arrays

Method	Description
transpose()	The transpose() method swaps the axes of the given array similar to the transpose of a matrix in mathematics
reshape()	The reshape() method array change the shape of an array without changing its data.
flatten()	The flatten() method flattens a NumPy array without changing its data.
resize()	resize() function is used to create a new array of different sizes and dimensions.

Broadcasting

- Broadcasting explains the way NumPy handles arrays of various shapes when performing arithmetic operations.
- Element-to-element operations cannot be performed on two arrays if their dimensions are different.
- NumPy still allows operations on arrays of different shape arrays due to the broadcasting feature.
- For the smaller array and the larger array to have similar shapes, they are broadcast to the same size.

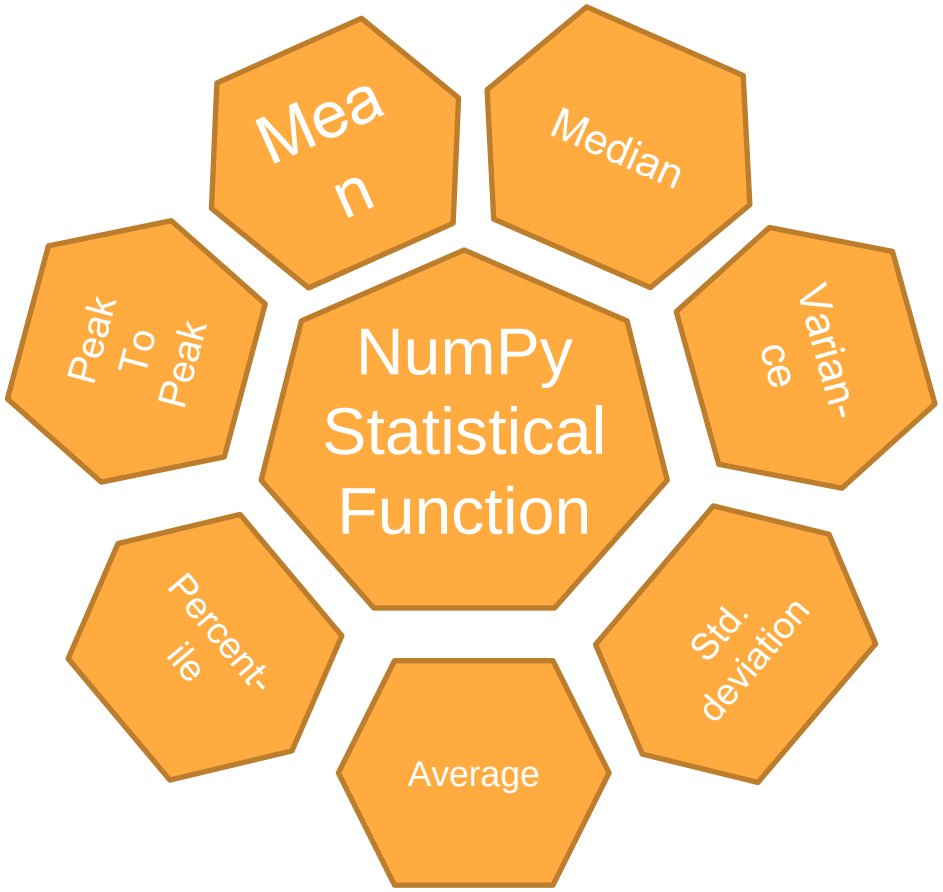


Source :

Reference

Statistical Functions

- Numpy includes a number of statistical functions that can be used to perform statistical data analysis.
- Statistics is concerned with gathering and analysing data.
- NumPy includes a number of statistical functions for analysing statistical data.



NumPy Statistical Functions

- **np.amin()**- Minimum value of the element along a specified axis.
- **np.amax()**- Maximum value of the element along a specified axis.
- **np.mean()**- Mean value of the data set.
- **np.median()**- Median value of the data set.
- **np.ptp()**- Range of values along an axis(peak to peak).
- **np.std()**- Standard deviation
- **np.var()** – Variance.
- **np.average()**- Weighted average
- **np.percentile()**- nth percentile of data along the specified axis.

Statistics - Standard Deviation and Variance

- **Standard deviation** is the square root of the average of squared deviations from mean. The function used for this is np.std().

  np.std([1,2,3,4])

- **Variance** is the average of squared deviations, i.e., **mean(abs(x - x.mean())**2).**

  Or, standard deviation is the square root of variance.

  np.var([1,2,3,4])

$$\text{Variance} = \sigma^2$$
$$= (\text{Standard Deviation})^2$$

$$\text{Standard Deviation} = \sigma$$
$$= \sqrt{(\text{Variance})}$$

Statistics - Mean, Median and Peak to Peak

- **Mean** - Compute the arithmetic mean along the specified axis.

  np.mean([1,2,3,4,5])

- **Median** - Compute the median along the specified axis.

  np.median([1,5,2,3,4])

- **PTP** - This function returns the range (maximum-minimum) of values along an axis.

  np.ptp(array)

I/O with NumPy

- The ndarray objects can be saved and loaded from disc files. NumPy offers input/output (I/O) functions for loading and saving data to and from files.
- Input/output functions support a variety of file formats, including binary and text formats.

Method	Description
save()	saves an array to a binary file in the NumPy .npy format.
load()	loads data from a binary file in the NumPy .npy format
savetxt()	saves an array to a text file in a specific format
loadtxt()	loads data from a text file.

I/O with NumPy

```
 import numpy as np
a = np.array([1,2,3,4,5])
np.save('outfile',a)
```

```
import numpy as np
b = np.load('outfile.npy')
print(b)
```

```
[1 2 3 4 5]
```

numpy.save()

- The input array is saved in a disc file with the npy extension by the numpy.save() function.

```
import numpy as np

a = np.array([1,2,3,4,5])
np.savetxt('out.txt',a)
b = np.loadtxt('out.txt')
print(b)
```

```
[1. 2. 3. 4. 5.]
```

savetxt()

- The savetxt() and loadtxt() functions are used to save and retrieve array data in simple text file format.

Lab - 1

Exploring NumPy for Data Manipulation

Conclusion

In this session we have learned -

- Introduction to NumPy library and where NumPy Library is used
- How we can download the NumPy Library
- What are NumPy Datatypes
- What is NumPy array and How to create a NumPy array.
- How we can perform I/O function with NumPy
- How Indexing , slicing and broadcasting function works in NumPy
- And last how we can perform the Statistical Function in NumPy

Source :

Reference

References

- https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)
- https://docs.python.org/3/library/
- https://www.tutorialspoint.com/numpy
- https://numpy.org/
- https://towardsdatascience.com/
- https://realpython.com/
- https://www.simplilearn.com/tutorials/python-tutorial/numpy-tutorial


Q&A
Let's Start

Quiz

1. **What does broadcasting in NumPy refer to?**

a) Broadcasting live events using NumPy

b) Transmitting data over the network using NumPy

c) Adjusting array shapes to perform element-wise operations

d) Generating random numbers with NumPy

Answer: C

Adjusting array shapes to perform element-wise operations

Quiz

2. Which function is used to calculate the standard deviation of an array in NumPy?

a) np.mean()

b) np.var()

c) np.std()

d) np.median()

Answer: C

np.std()

Quiz

3. What is the main advantage of using NumPy arrays over Python lists for numerical computations?

a) NumPy arrays are more memory efficient

b) Python lists support element-wise operations

c) NumPy arrays can hold elements of different data types

d) Python lists are faster for large datasets

Answer: A

NumPy arrays are more memory efficient

Quiz

4. Which of the following is true about ‘np.savetxt()’ ?

a) It reads data from a CSV file.

b) It reads data from a binary file.

c) It writes data to a text file.

d) writes data to a binary file.

Answer: C

It writes data to a text file.

Quiz

5. In NumPy, what does slicing refer to?

a) Dividing an array into equal parts.

b) Extracting a subset of elements from an array.

c) Rearranging the elements of an array.

d) Combining two arrays into one.

Answer: B

Extracting a subset of elements from an array.

Thank You