Bài 4:
LẬP TRÌNH PYTHON CƠ BẢN
(Sử dụng thư viện NumPy làm việc với ma trận - 01)

AI Academy Vietnam

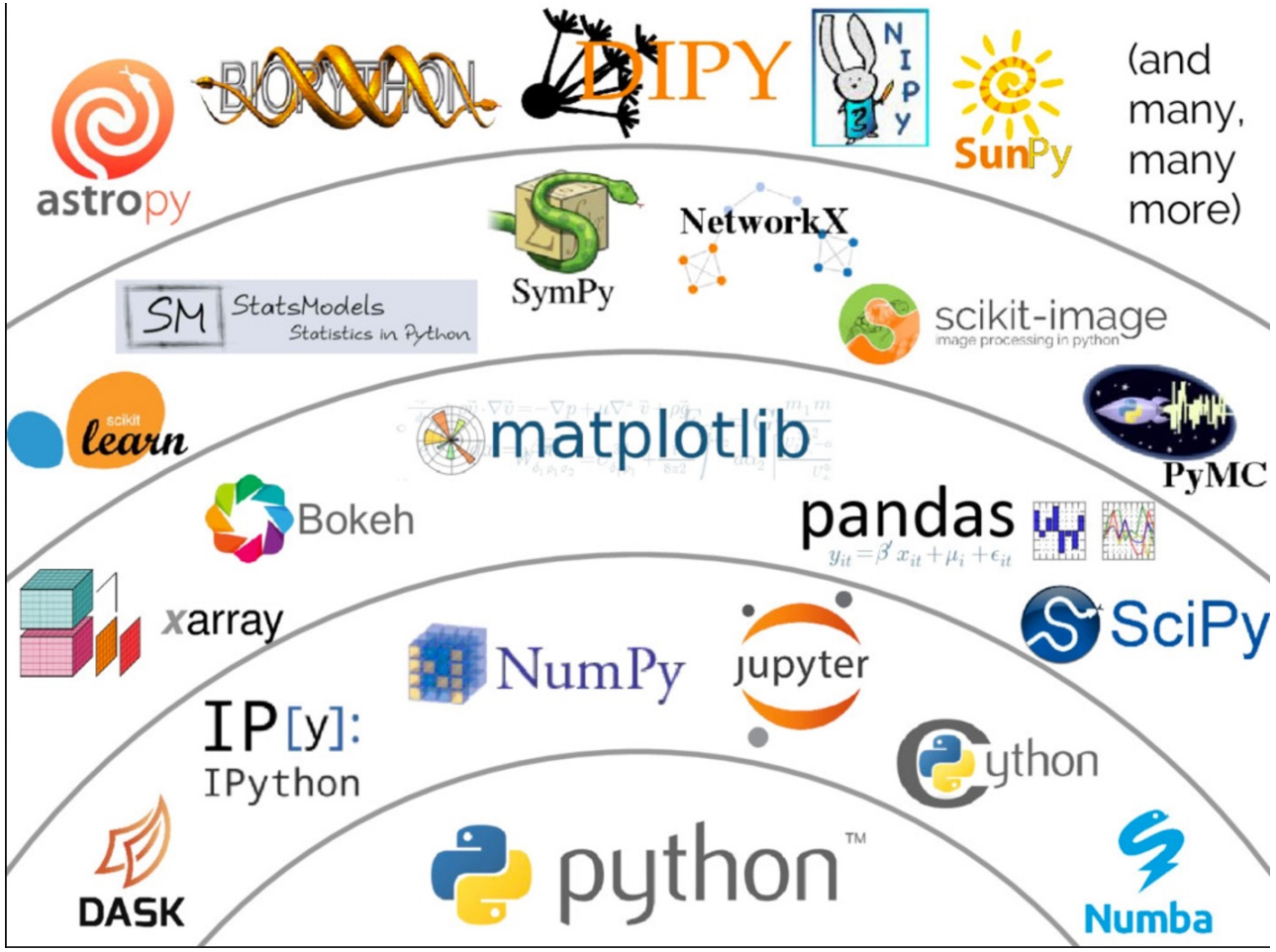
Nội dung bài 4

1. Một số thư viện quan trọng của Python
2. Giới thiệu thư viện Numpy
3. Cách tạo vector, ma trận (matrix)
   - Tạo mảng 1D, 2D, 3D | Tạo mảng với các hàm có sẵn | Tạo mảng từ file dữ liệu
4. Thao tác cơ bản với mảng
   - Quan sát thuộc tính | Chuyển đổi kiểu dữ liệu | Truy cập phần tử
5. Tính toán các đặc trưng thống kê
   - Min, Max, Mean, Median, Mode, Range, Std, Corrcoef

1. Một số thư viện quan trong của Python

Thư viện Python

- **Python** có hệ thống thư viện rất phong phú, hỗ trợ nhiều lĩnh vực khác nhau.



- Do đó, tùy thuộc vào lĩnh vực nghiên cứu cụ thể, để lựa chọn và sử dụng các thư viện cho phù hợp.

Thư viện Python

Python libraries for Data Analysis

There are many interesting libraries that have made Python popular with Data Scientists:

pandas

NumPy

matplotlib

Top 10 Python Packages 2020

Data Science in Python

Pandas,

Scikit-learn, Numpy

Matplotlib

Top 5 Python Libraries for Data Science

python

Report from Cloud Academy suggests that the top technical skill in demand for data engineers is python. 67 percent of job posts mentioned python.

1.) NUMPY

NumPy

Through NumPy, you can use it as an efficient multi-dimensional container of generic data. It also contains sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities

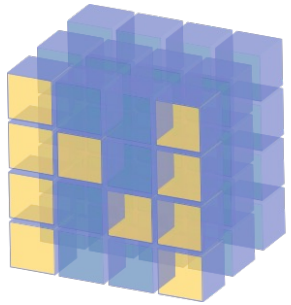
2.) PANDAS

Pandas provides high-performance, easy-to-use data structures and data analysis tools for python. You can store and manage data from tables by performing manipulation over rows and columns.

3.) SCIKIT-LEARN

Scikit-Learn is a powerful library for machine learning in Python. It contains simple and efficient tools for data mining and data analysis. It is built on NumPy, SciPy and matplotlib.

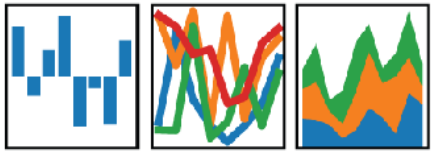
Thư viện Python



NumPy

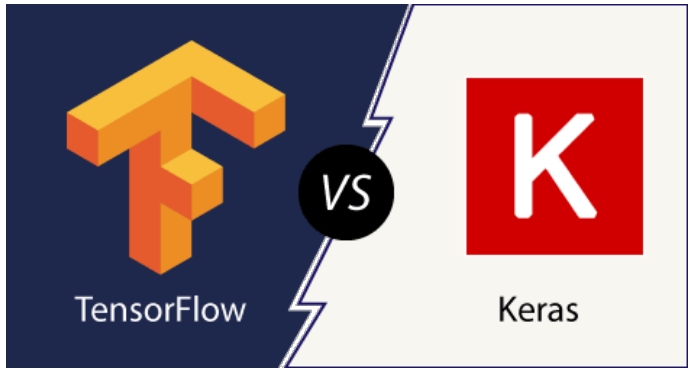
pandas

$$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$$



	BandName	WavelengthMax	WavelengthMin
0	CoastalAerosol	450	430
1	Blue	510	450
2	Green	590	530
3	Red	670	640
4	NearInfrared	880	850
5	ShortWaveInfrared_1	1650	1570
6	ShortWaveInfrared_2	2290	2110
7	Cirrus	1380	1360

Machine Learning with Scikit-Learn



Natural Language Analyses with NLTK

Thư viện Python

- Khai báo sử dụng thư viện trong Python

In [1]:
```
#Khai báo sử dụng thư viện và kiểm tra phiên bản thư viện đang sử dụng
import numpy as np
print("Thu vien Numpy, Version: ",np.__version__)
```

```
Thu vien Numpy, Version:  1.15.4
```

In [2]:
```
#Trong trường hợp thư viện chưa được cài đặt!
import scrapy as sc
print("Thu vien Scrapy, Version: ",sc.__version__)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-2-ef1be0ed66f4> in <module>
      1 #Trong trường hợp thư viện chưa được cài đặt!
----> 2 import scrapy as sc
      3 print("Thu vien Scrapy, Version: ",sc.__version__)

ModuleNotFoundError: No module named 'scrapy'
```

2. Thư viện NumPy

Thư viện Numpy

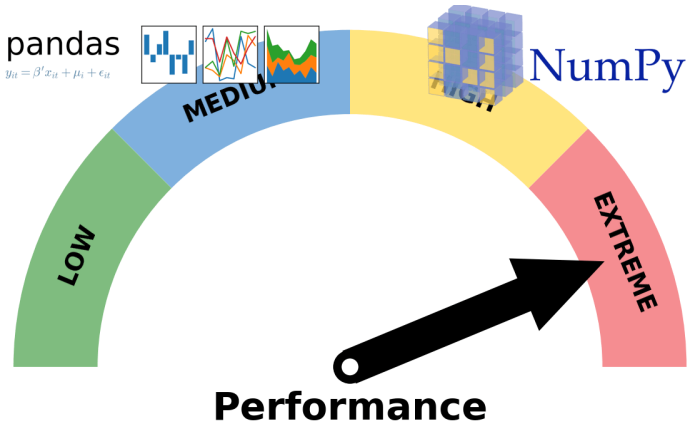
- **Numpy** (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python.
- Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng "core Python" đơn thuần.
- Ngoài ra, Python cũng hỗ trợ một thư viện khác để mở rộng thêm các tính năng của Numpy là Scipy với ưu thế về các phép hồi quy hay biến đổi Fourier…
- Tham khảo thêm tại: http://www.numpy.org/

```
1 big_array = np.random.rand(1000000)
2 %timeit sum(big_array)
3 %timeit np.sum(big_array)
```

```
10 loops, best of 3: 171 ms per loop
1000 loops, best of 3: 380 µs per loop
```

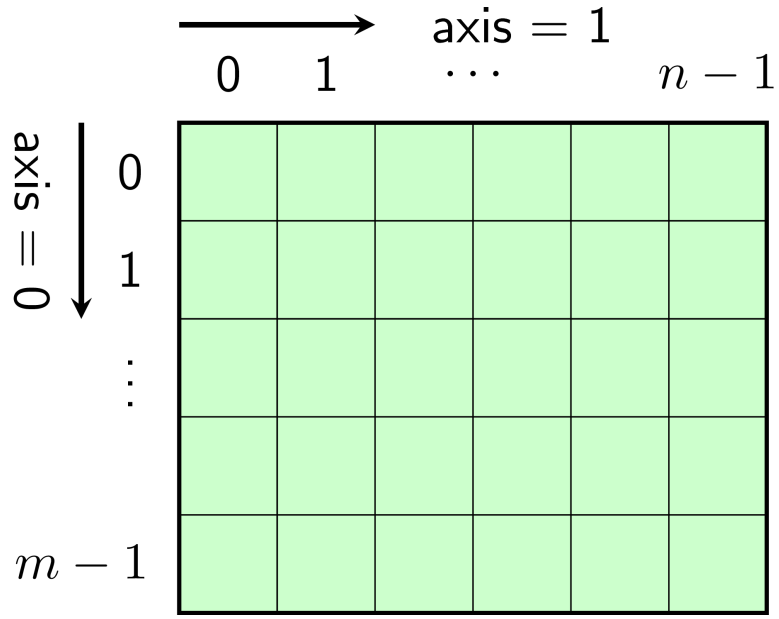
```
1 %timeit min(big_array)
2 %timeit np.min(big_array)
```

```
10 loops, best of 3: 103 ms per loop
1000 loops, best of 3: 432 µs per loop
```



Thư viện Numpy

- **Đối tượng chính của NumPy** là các mảng đa chiều đồng nhất:
  - Kiểu dữ liệu của các phần tử con trong mảng phải **giống nhau**
  - Mảng có thể có 1 chiều hoặc nhiều chiều
  - Các chiều được đánh số từ 0 trở đi
  - Số chiều được gọi là hạng (**rank**)
  - Có đến 24 kiểu số khác nhau.
  - Kiểu ndarray là lớp chính xử lý dữ liệu mảng nhiều chiều.
  - Có rất nhiều hàm và phương thức xử lý mảng



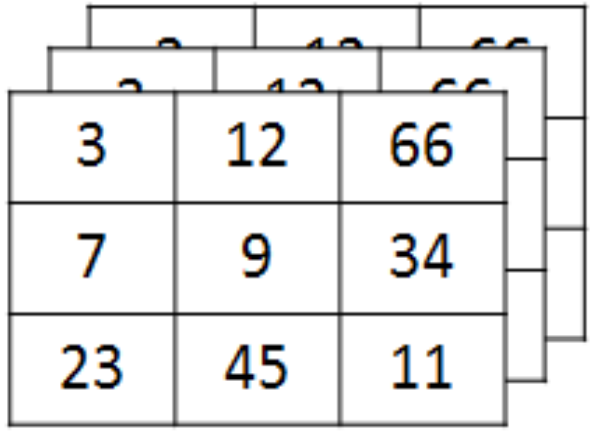
Thư viện Numpy

1	5	18	23

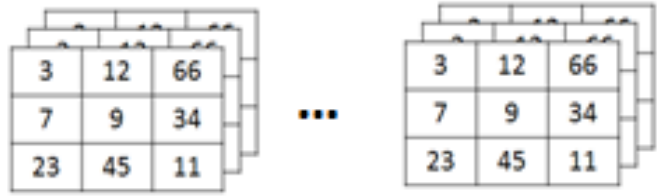
Vector (1D array)
Dimension = 1
(1 index required)

3	12	66
7	9	34
23	45	11

Matrix (2D array)
Dimension = 2
(2 indexes required)



3D array (3rd order Tensor)
Dimension = 3
(3 indexes required)



ND array
Dimension = N
(N indexes required)

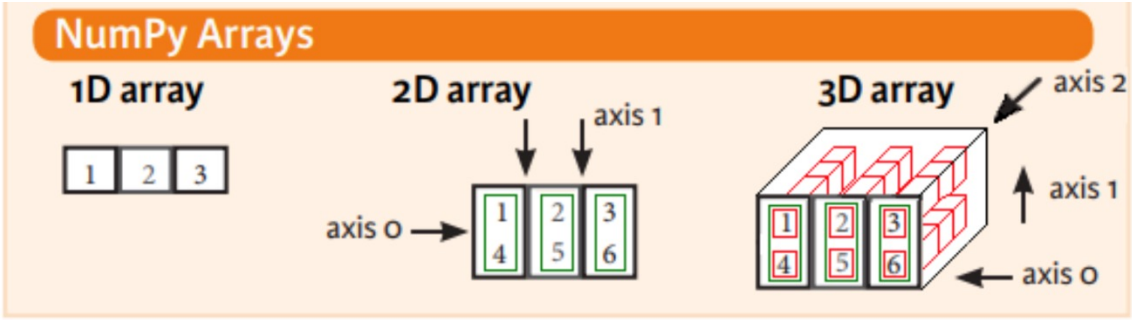
3. Khởi tạo mảng

Khởi tạo mảng

Khởi tạo mảng 1 chiều – 1D (Vector)

```python
#Khởi tạo mảng 1 chiều với thư viện Numpy
import numpy as np

#Tạo mảng 1 chiều (1D) - row
a = np.array([1, 2, 5, 7, 0, 8])

print(a)
print("Loại dữ liệu của biến a:", type(a))
print("Kiểu dữ liệu của phần tử trong mảng a:", a.dtype)
print("Kích thước của mảng a:", a.shape)
print("Số phần tử của mảng a:", a.size)
print("Số chiều của mảng a:", a.ndim)
```

```
[1 2 5 7 0 8]
Loại dữ liệu của biến a: <class 'numpy.ndarray'>
Kiểu dữ liệu của phần tử trong mảng a: int32
Kích thước của mảng a: (6,)
Số phần tử của mảng a: 6
Số chiều của mảng a: 1
```

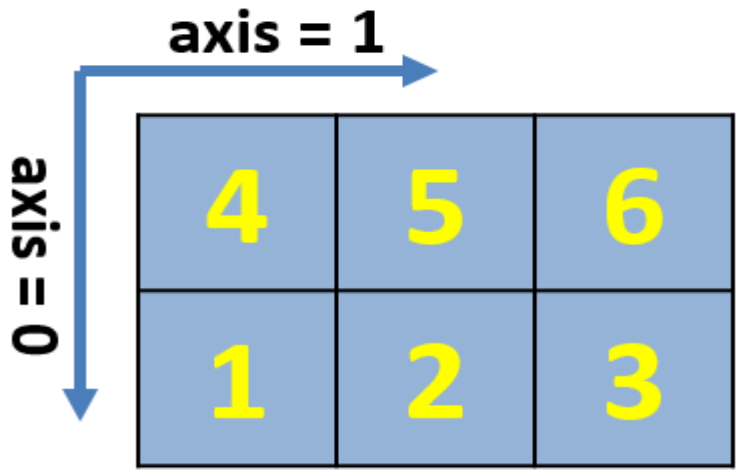


Khởi tạo mảng (2)

- **Khởi tạo mảng 2 chiều – 2D (Matrix)**

```python
#Gọi thư viện numpy
import numpy as np

#Tạo mảng 2 chiều (2D - Ma trận)
b = np.array([(4, 5, 6.0),(1, 2, 3.5)])

print(b)
print("Loại dữ liệu của biến b:", type(b))
print("Kiểu dữ liệu của phần tử trong mảng b:", b.dtype)
print("Kích thước của mảng b:", b.shape)
print("Số phần tử của mảng b:", b.size)
print("Số chiều của mảng b:", b.ndim)
```

```
[[4.  5.  6. ]
 [1.  2.  3.5]]
Loại dữ liệu của biến b: <class 'numpy.ndarray'>
Kiểu dữ liệu của phần tử trong mảng b: float64
Kích thước của mảng b: (2, 3)
Số phần tử của mảng b: 6
Số chiều của mảng b: 2
```

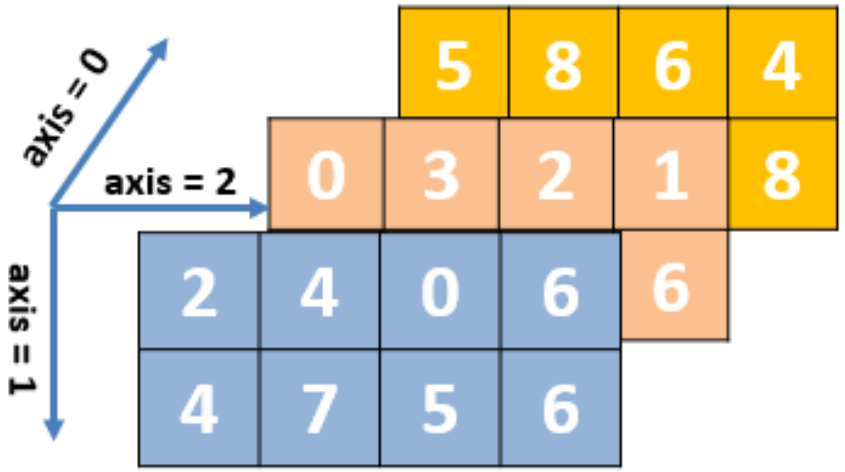


Khởi tạo mảng (3)

- **Khởi tạo mảng 3 chiều – 3D**

```python
import numpy as np

c = np.array([[(2,4,0,6), (4,7,5,6)],
              [(0,3,2,1), (9,4,5,6)],
              [(5,8,6,4), (1,4,6,8)]]) #mảng 3 chiều (3D)

print(c)
print("Phần tử đầu tiên của mảng c:",c[0,0,0])
print("Kiểu dữ liệu của phần tử trong mảng c:",c.dtype)
print("Kích thước của mảng c:",c.shape)
print("Số phần tử của mảng c:",c.size)
print("Số chiều của mảng c:",c.ndim)
```

```
[[[2 4 0 6]
  [4 7 5 6]]

 [[0 3 2 1]
  [9 4 5 6]]

 [[5 8 6 4]
  [1 4 6 8]]]
Phần tử đầu tiên của mảng c: 2
Kiểu dữ liệu của phần tử trong mảng c: int32
Kích thước của mảng c: (3, 2, 4)
Số phần tử của mảng c: 24
Số chiều của mảng c: 3
```



Khởi tạo mảng
Với các hàm có sẵn của NumPy

Khởi tạo mảng (4)

- **Khởi tạo mảng với các hàm sẵn có của Numpy**

Initial Placeholders

`>>> np.zeros((3,4))`	Create an array of zeros
`>>> np.ones((2,3,4),dtype=np.int16)`	Create an array of ones
`>>> d = np.arange(10,25,5)`	Create an array of evenly spaced values (step value)
`>>> np.linspace(0,2,9)`	Create an array of evenly spaced values (number of samples)
`>>> e = np.full((2,2),7)`	Create a constant array
`>>> f = np.eye(2)`	Create a 2X2 identity matrix
`>>> np.random.random((2,2))`	Create an array with random values
`>>> np.empty((3,2))`	Create an empty array

Khởi tạo mảng (5)

- **Vd 1: Tạo ma trận 0|1 kích thước m x n**

```python
# Phương thức zeros: Tạo ma trận 0 kích thước 5 hàng x 3 cột
import numpy as np

array_zeros = np.zeros((5, 3))

print(array_zeros)
print("Kiểu dữ liệu trong mảng array_zeros:", array_zeros.dtype)
print("Kích thước của mảng array_zeros:", array_zeros.shape)
print("Số phần tử của mảng array_zeros:", array_zeros.size)
print("Số chiều của mảng array_zeros:", array_zeros.ndim)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
Kiểu dữ liệu trong mảng array_zeros: float64
Kích thước của mảng array_zeros: (5, 3)
Số phần tử của mảng array_zeros: 15
Số chiều của mảng array_zeros: 2
```

```python
# Phương thức ones: Tạo ma trận 1 kích thước 3 hàng x 5 cột
import numpy as np

array_one = np.ones((3, 5),dtype=np.int)

print(array_one)
print("Kiểu dữ liệu trong mảng array_one:", array_one.dtype)
print("Kích thước của mảng array_one:", array_one.shape)
print("Số phần tử của mảng array_one:", array_one.size)
print("Số chiều của mảng array_one:", array_one.ndim)
```

```
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]]
Kiểu dữ liệu trong mảng array_one: int32
Kích thước của mảng array_one: (3, 5)
Số phần tử của mảng array_one: 15
Số chiều của mảng array_one: 2
```

Khởi tạo mảng (6)

- **Vd 2: Tạo ma trận đơn vị cập n**

```python
#Phương thức eye: Tạo ma trận đơn vị cấp 5
import numpy as np
array_eye = np.eye(5)

print(array_eye)
print("Kiểu dữ liệu của phần tử trong mảng array_eye:", array_eye.dtype)
print("Kích thước của mảng array_eye:", array_eye.shape)
print("Số phần tử của mảng array_eye:", array_eye.size)
print("Số chiều của mảng array_eye:", array_eye.ndim)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
Kiểu dữ liệu của phần tử trong mảng array_eye: float64
Kích thước của mảng array_eye: (5, 5)
Số phần tử của mảng array_eye: 25
Số chiều của mảng array_eye: 2
```

Khởi tạo mảng (7)

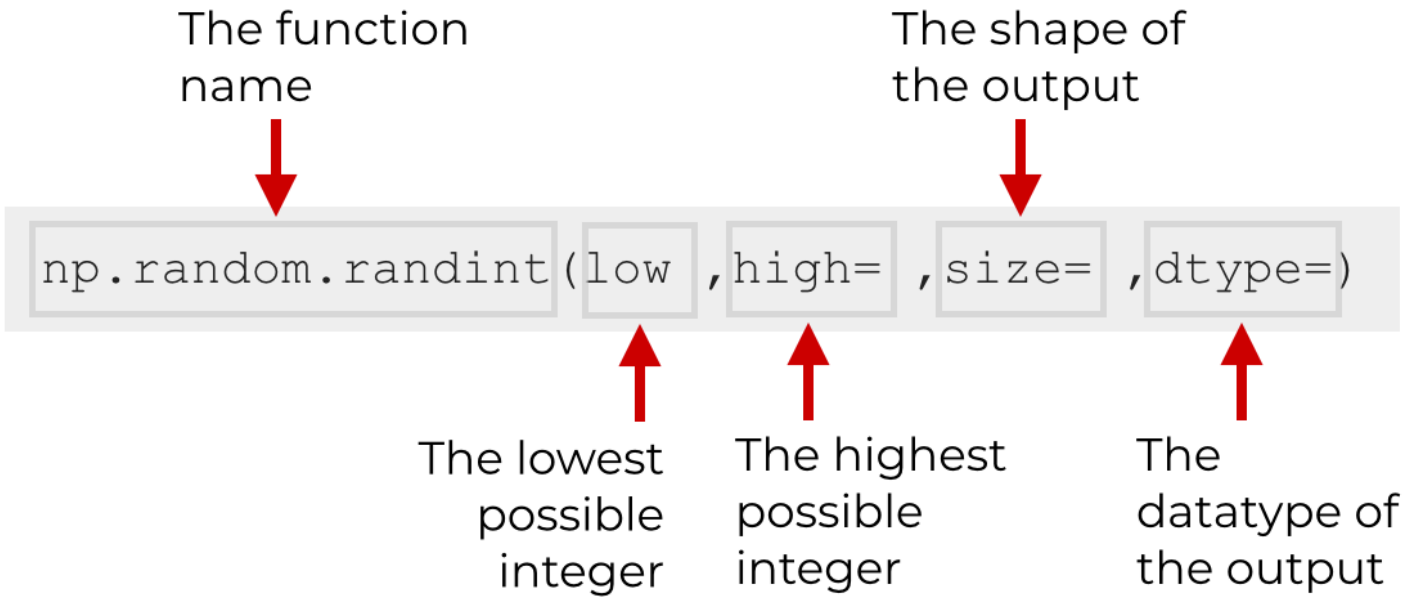
- **Vd 3: Tạo ma trận với các phần tử ngẫu nhiên trong khoảng (0,1)**

```python
#Phương thức random: Tạo một ma trận (7x5) các phần tử ngẫu nhiên [0,1]
import numpy as np
array_random = np.random.random((7,5))

print(array_random)
print("Kiểu dữ liệu của phần tử trong mảng array_random:", array_random.dtype)
print("Kích thước của mảng array_random:", array_random.shape)
print("Số phần tử của mảng array_random:", array_random.size)
print("Số chiều của mảng array_random:", array_random.ndim)
```

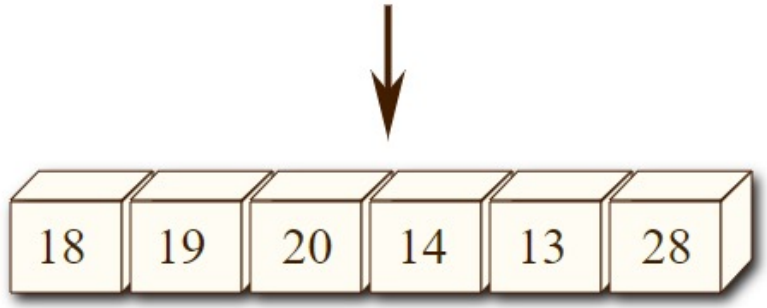
```
[[0.57738653 0.38330643 0.84085595 0.88920867 0.11759141]
 [0.13200344 0.40891213 0.46518628 0.81332657 0.62117097]
 [0.0255157  0.10842881 0.36001561 0.06382023 0.40403947]
 [0.37338483 0.35678386 0.38280971 0.97395415 0.8950108 ]
 [0.86054013 0.93742679 0.13039088 0.599897   0.0071806 ]
 [0.83131566 0.35010989 0.47524521 0.56107776 0.13418245]
 [0.39089618 0.40229334 0.73431802 0.53481456 0.45046071]]
Kiểu dữ liệu của phần tử trong mảng array_random: float64
Kích thước của mảng array_random: (7, 5)
Số phần tử của mảng array_random: 35
Số chiều của mảng array_random: 2
```

Khởi tạo mảng (8)

- **Vd 4: Tạo vector, ma trận với các phần tử là số nguyên ngẫu nhiên trong khoảng (low,high)**



np.random.randint(low=10, high=30, size=6)

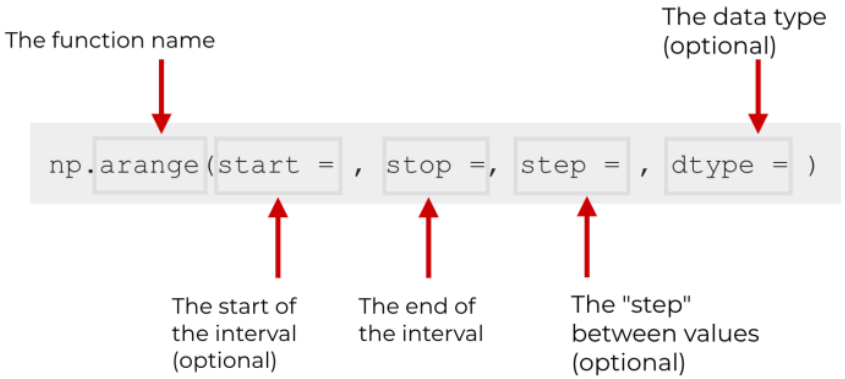


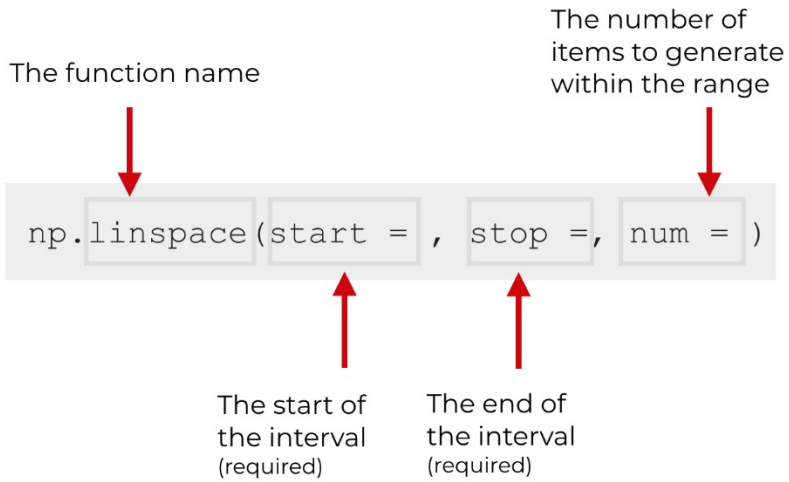
Khởi tạo mảng (9)

- **Vd 5: Tạo vector với các tham số thiết lập**

```
1  #Phương thức arange(a, b, steps):
2  #Tạo vector:
3  # Phần tử đầu tiên = a,
4  # kết thúc <b,
5  # mỗi phần tử cách nhau một khoảng = steps
6  d = np.arange(1, 15, 2)
7  print('Vector d:', d)
8  print("Số phần tử của vector d:", d.size)
9
10 print('------------------------------------------------------------')
11 #Phương thức linspace(a, b, num)
12 #Tạo vector:
13 #Phần tử đầu tiên  = a,
14 #Phần tử kết thúc  = b,
15 #Số phần tử của ma trận = num
16 f = np.linspace(1,15,11)
17 print('Vector f:', f)
18 print("Số phần tử của vector f:", f.size)
19
```

```
Vector d: [ 1  3  5  7  9 11 13]
Số phần tử của vector d: 7
------------------------------------------------------------
Vector f: [ 1.   2.4  3.8  5.2  6.6  8.   9.4 10.8 12.2 13.6 15. ]
Số phần tử của vector f: 11
```





Khởi tạo mảng

từ file dữ liệu .txt

Khởi tạo mảng (10)

- **Bảng điểm của lớp 2A** (bao gồm 30 học sinh, tương ứng với 30 cột, của 10 môn học, tương ứng với 10 hàng) lưu trong file Diem_2A.txt

Khởi tạo mảng (11)

- **Đọc dữ liệu từ file txt vào biến mảng.**

```python
import numpy as np

#Đọc dữ liệu từ file Diem_2A.txt
path  ='Data_Excercise\Diem_2A.txt'
diem_2a = np.loadtxt(path,delimiter=',',dtype=np.int)

print(diem_2a)
print("Kiểu dữ liệu của phần tử trong mảng diem_2a:", diem_2a.dtype)
print("Kích thước của mảng diem_2a:", diem_2a.shape)
print("Số phần tử của mảng diem_2a:", diem_2a.size)
print("Số chiều của mảng diem_2a:", diem_2a.ndim)
```

```
[[ 2  4  3  7  5  6  5  6  8  9  3  6  1  9  8  7  3  3  9  5  1  6  5  1
   4  6  7  1  1  1]
 [ 3  5  3 10  9  1  9  8  3  1  6  0  7 10  8  5  2  7  7  1  1  6  1  6
   3  0  2  2  1  6]

 [ 6  7  8  9 10  9  2  2  6  1 10  9  6  3  9  5  9  8  1  1  8  8  8  6
   6  8  7  3  8  1]
 [ 7  8  7  8  6 10 10  6  8 10  8  9  8  8  5 10  8  7  8  7  9  9  8  7
   7  7 10  8  9  7]]
Kiểu dữ liệu của phần tử trong mảng diem_2a: int32
Kích thước của mảng diem_2a: (10, 30)
Số phần tử của mảng diem_2a: 300
Số chiều của mảng diem_2a: 2
```

4. Các thao tác cơ bản

4.1 Quan sát mảng

```
#a.shape: Cho biết kích thước của mảng a:
print('kích thước của mảng diem_2a:', diem_2a.shape)
```

```
kích thước của mảng diem_2a: (10, 30)
```

```
#a.ndim: Cho biết Số chiều của mảng a:
print('Số chiều của mảng diem_2a:', diem_2a.ndim)
```

```
Số chiều của mảng diem_2a: 2
```

```
#a.size: Cho biết số phần tử của mảng a:
print('Số phần tử của mảng diem_2a: ', diem_2a.size)
```

```
Số phần tử của mảng diem_2a:  300
```

```
#a.dtype: Cho biết kiểu dữ liệu của các phần tử trong mảng a
print('Kiểu dữ liệu của các phần tử trong mảng diem_2a:', diem_2a.dtype)
```

```
Kiểu dữ liệu của các phần tử trong mảng diem_2a: float64
```

4.2 Chuyển đổi kiểu dữ liệu

```python
#a.astype(kiểu mới): Chuyển đổi kiểu dữ liệu của các phần tử
a_float = np.linspace(0,15,11)
print(a_float)
print('Kiểu Dữ liệu: ', a_float.dtype)
print('----------------------------------------')
#Chuyển từ kiểu float --> int
a_int = a_float.astype(np.int16)
print(a_int)
print('Dữ liệu sau khi chuyển: ', a_int.dtype)
```

```
[ 0.   1.5  3.   4.5  6.   7.5  9.  10.5 12.  13.5 15. ]
Kiểu Dữ liệu:  float64
----------------------------------------
[ 0  1  3  4  6  7  9 10 12 13 15]
Dữ liệu sau khi chuyển:  int16
```

```python
#Chuyển từ kiểu float --> int
a_str = a_int.astype(np.str)
print(a_str)
print('Dữ liệu sau khi chuyển: ', a_str.dtype)
print('----------------------------------------')
#Chuyển từ kiểu float --> boolean
a_bol = a_int.astype(np.bool)
print(a_bol)
print('Dữ liệu sau khi chuyển: ', a_bol.dtype)
```

```
['0' '1' '3' '4' '6' '7' '9' '10' '12' '13' '15']
Dữ liệu sau khi chuyển:  <U6
----------------------------------------
[False  True  True  True  True  True  True  True  True  True  True]
Dữ liệu sau khi chuyển:  bool
```

Chuyển đổi kiểu dữ liệu (2)

NumPy dtypes

ENTHOUGHT

Basic Type	Available NumPy types	Comments
Boolean	bool	Elements are 1 byte in size
Integer	int8, int16, int32, int64, int128, int	int defaults to the size of int in C for the platform
Unsigned Integer	uint8, uint16, uint32, uint64, uint128, uint	uint defaults to the size of unsigned int in C for the platform
Float	float32, float64, float, longfloat,	Float is always a double precision floating point value (64 bits). longfloat represents large precision floats. Its size is platform dependent.
Complex	complex64, complex128, complex	The real and complex elements of a complex64 are each represented by a single precision (32 bit) value for a total size of 64 bits.
Strings	str, unicode	Unicode is always UTF32 (UCS4)
Object	object	Represent items in array as Python objects.
Records	void	Used for arbitrary data structures in record arrays.

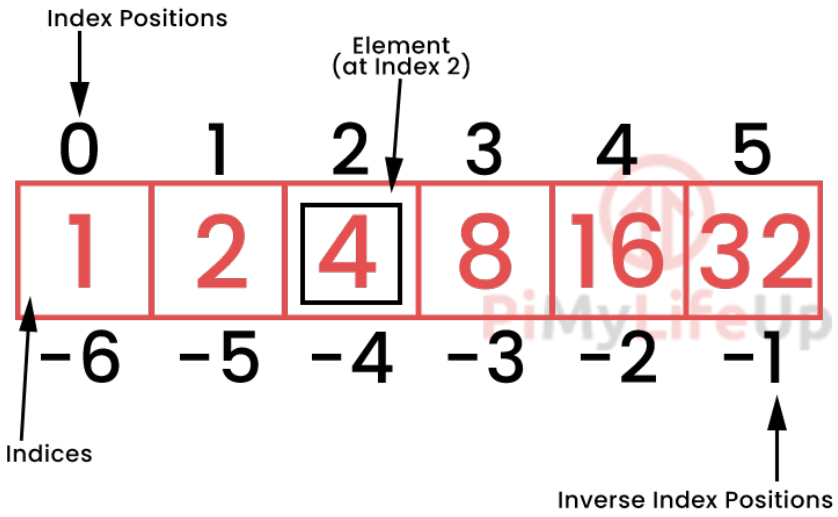
18

Chuyển đổi kiểu dữ liệu (3)

Data type	Description
`bool_`	Boolean (True or False) stored as a byte
`int_`	Default integer type (same as C `long`; normally either `int64` or `int32`)
`intc`	Identical to C `int` (normally `int32` or `int64`)
`intp`	Integer used for indexing (same as C `ssize_t`; normally either `int32` or `int64`)
`int8`	Byte (-128 to 127)
`int16`	Integer (-32768 to 32767)
`int32`	Integer (-2147483648 to 2147483647)
`int64`	Integer (-9223372036854775808 to 9223372036854775807)
`uint8`	Unsigned integer (0 to 255)
`uint16`	Unsigned integer (0 to 65535)
`uint32`	Unsigned integer (0 to 4294967295)
`uint64`	Unsigned integer (0 to 18446744073709551615)
`float_`	Shorthand for `float64`.
`float16`	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
`float32`	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
`float64`	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
`complex_`	Shorthand for `complex128`.
`complex64`	Complex number, represented by two 32-bit floats
`complex128`	Complex number, represented by two 64-bit floats

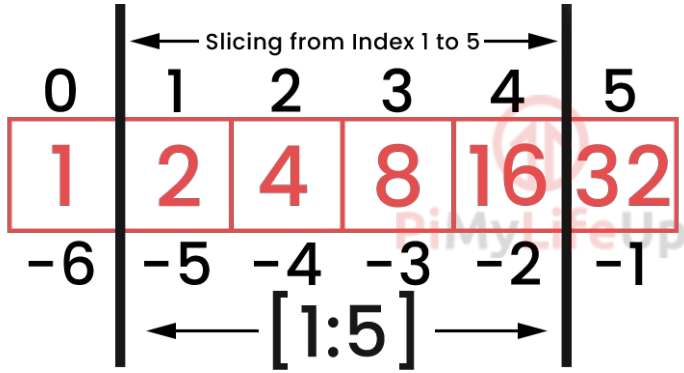
4.3 Truy cập tới các phần tử

- Truy cập tới phần tử trong một vector (1D)




```
#Truy cập tới một phần tử của Vector: a[index]
#Note: index phần tử đầu tiên 0
#    : index phần tử cuối cùng -1
a = np.array([3,  5,  3, 10,  9,  1,  9,  8,  3,  1])

print('các phần tử của Vector a:\n', a)
print('-----------------------------------------------')
print('phần tử đầu tiên:', a[0])
print('phần tử thứ 3:', a[3])
print('phần tử cuối cùng:', a[-1])
```

```
các phần tử của Vector a:
 [ 3  5  3 10  9  1  9  8  3  1]
-----------------------------------------------
phần tử đầu tiên: 3
phần tử thứ 3: 10
phần tử cuối cùng: 1
```




```
#Truy cập tới nhiều phần tử của Vector: a[index1:index2]
print('các phần tử của Vector a:\n', a)
print('-----------------------------------------------')
print('3 Phần tử đầu tiên:', a[:3])
print('Từ phần tử thứ 5 tới hết:', a[5:])
print('Từ phần tử 2 đến phần tử <6 của vector:',a[2:6])
```

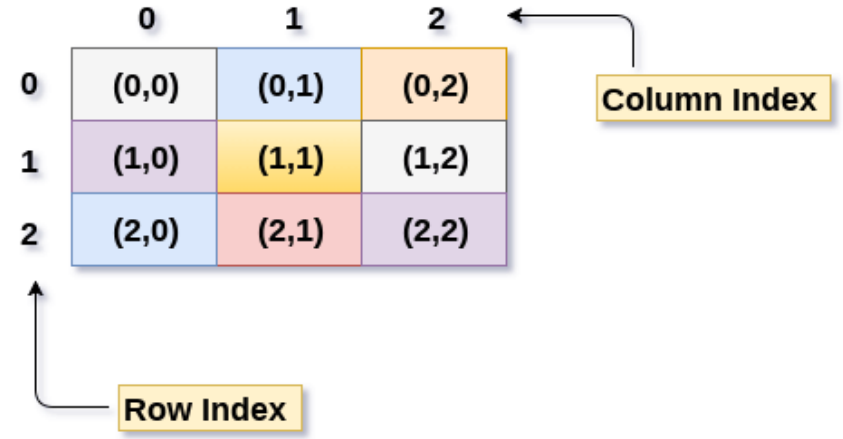
```
các phần tử của Vector a:
 [ 3  5  3 10  9  1  9  8  3  1]
-----------------------------------------------
3 Phần tử đầu tiên: [3 5 3]
Từ phần tử thứ 5 tới hết: [1 9 8 3 1]
Từ phần tử 2 đến phần tử <6 của vector: [ 3 10  9  1]
```

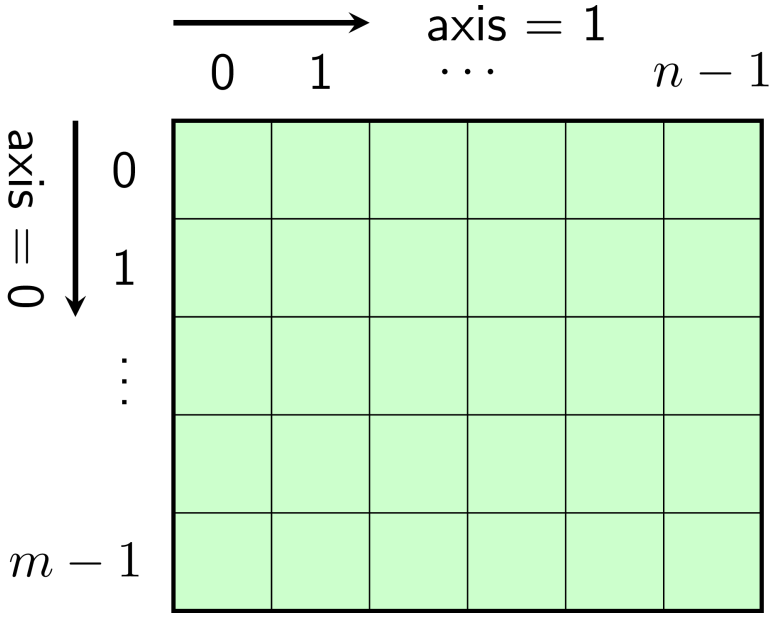
Truy cập tới các phần tử (2)

- Truy cập tới các phần tử trong một ma trận (2D)

```
1  #Truy cập tới 1 phần tử của ma trận (2D): a[index_row, index_col]
2  print('Điểm môn học đầu tiên, của học sinh đầu tiên:',diem_2a[0,0])
3  print('Điểm môn học thứ 1, của học sinh thứ 3:',diem_2a[1,3])
4  print('Điểm môn cuối cùng, của học sinh cuối cùng:',diem_2a[-1,-1])
5  print('-------------------------------------------------------')
6  print('Bảng điểm lớp 2A:\n',diem_2a)
```

```
Điểm môn học đầu tiên, của học sinh đầu tiên: 2
Điểm môn học thứ 1, của học sinh thứ 3: 10
Điểm môn cuối cùng, của học sinh cuối cùng: 7
```





Truy cập tới các phần tử (3)

- Truy cập tới các phần tử trong một ma trận (2D)

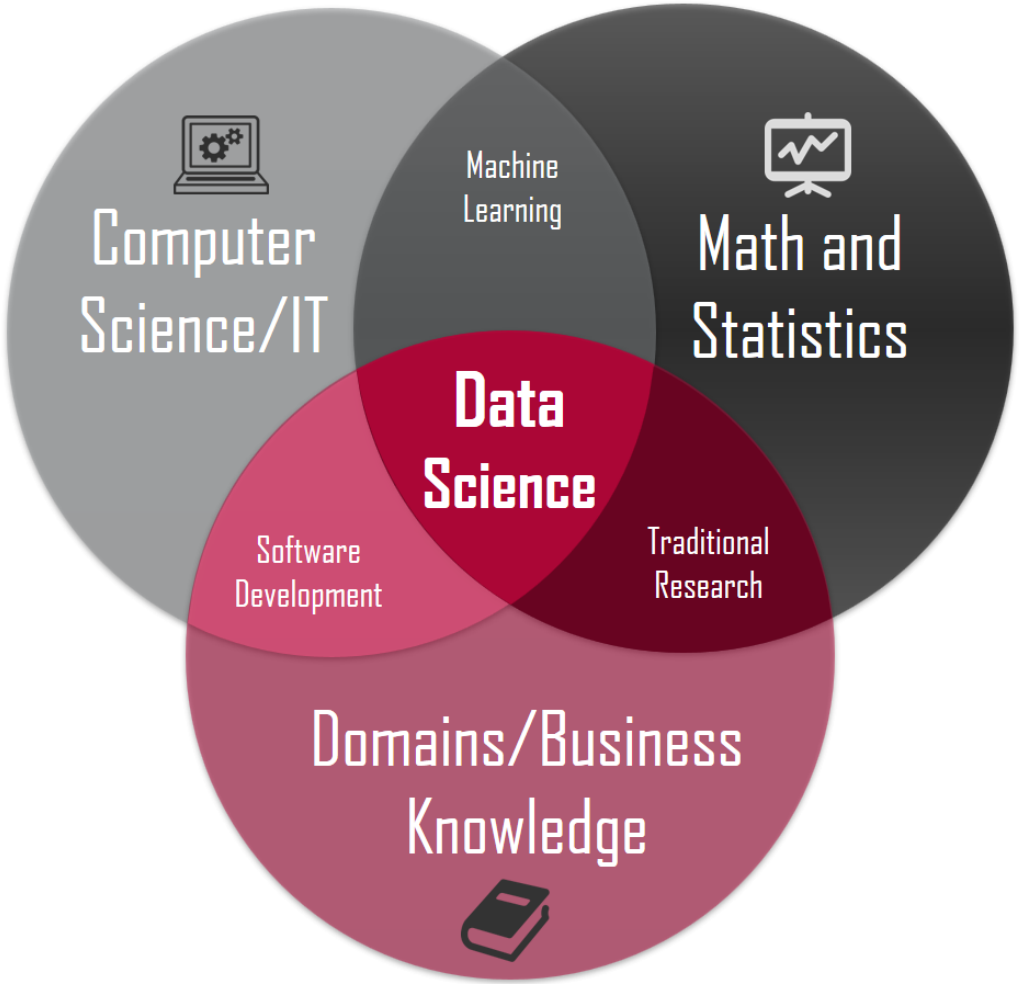
```
#Truy cập tới nhiều phần tử trong ma trận: a[index_row1:index_row2,index_col1:index_col2]
#Lấy điểm tất cả các môn (tất cả các hàng) của học sinh 5:
diem_hs5 = diem_2a[:,5]
print("Điểm các môn của học sinh 5:",diem_hs5)

#Lấy điểm môn học cuối cùng của tất cả học sinh (tất cả các cột)
diem_mon = diem_2a[-1,:]
print("Điểm môn học cuối cùng của tất cả học sinh: \n",diem_mon)

#Lấy điểm 5 môn học đầu tiên của 10 học sinh đầu tiên
diem5_hs10 = diem_2a[:5,:10]
print("Bảng điểm 5 môn học đầu tiên của 10 học sinh đầu của lớp:\n",diem5_hs10)
```

```
Điểm các môn của học sinh 5: [ 6  1  9  7  5  5  9  7  9 10]
Điểm môn học cuối cùng của tất cả học sinh:
 [ 7  8  7  8  6 10 10  6  8 10  8  9  8  8  5 10  8  7  8  7  9  9  8  7
  7  7 10  8  9  7]
Bảng điểm 5 môn học đầu tiên của 10 học sinh đầu của lớp:
 [[ 2  4  3  7  5  6  5  6  8  9]
 [ 3  5  3 10  9  1  9  8  3  1]
 [ 1 10  4  9  6  9  0  2  3  1]
 [ 6  3  0  8  3  7  7  2  6  8]
 [ 4  3  6  7  4  5  2  6  9  4]]
```

THỰC HÀNH 1

5. Tính toán các đặc trưng thống kê

5. Các đặc trưng thống kê



Toán học và thống kê có một vai trò rất quan trọng trong khoa học dữ liệu!

5.1 Max - Min

- a.max(): Lấy giá trị lớn nhất của mảng a
- b.min(): Lấy giá trị nhỏ nhất của mảng b

```
#Max - Min: Xác định giá trị lớn nhất, nhỏ nhất:
#1) Hiển thị điểm cao nhất, thấp nhất của lớp 2A
print('Điểm cao nhất của lớp:',diem_2a.max())
print('Điểm thấp nhất của lớp:',diem_2a.min())
```

```
Điểm cao nhất của lớp: 10
Điểm thấp nhất của lớp: 0
```

```
#2) Liệt kê điểm cao nhất và thấp nhất theo môn học
for i in range(0,diem_2a.shape[0]):
  print('Môn ', i,': Điểm Max: ', diem_2a[i,:].max(),
                '-- Điểm Min:',diem_2a[i,:].min())
```

```
Môn  0 : Điểm Max:  9 -- Điểm Min: 1
Môn  1 : Điểm Max:  10 -- Điểm Min: 0
Môn  2 : Điểm Max:  10 -- Điểm Min: 0
```

```
#3) Liệt kê điểm cao nhất và thấp nhất của mỗi học sinh
for i in range(0,diem_2a.shape[1]):
  print('Học sinh ', i,': Điểm Max: ', diem_2a[:,i].max(),
                      '-- Điểm Min:',diem_2a[:,i].min())
```

```
Học sinh  0 : Điểm Max:  9 -- Điểm Min: 1
Học sinh  1 : Điểm Max:  10 -- Điểm Min: 3
```

5.2 Sum

- **a.sum():Tính tổng tất cả các phần tử của mảng a**

```
#Sum:Tính tổng các phần tử trong mảng
print('Tổng tất các điểm trong của lớp 2A:', diem_2a.sum())
print('-------------------------------------------------')

#Tính tổng điểm của từng học sinh:
for i in range(0,diem_2a.shape[1]):
  print('Tổng điểm các môn của học sinh ', i,' : ', diem_2a[:,i].sum())
```

```
Tổng tất các điểm trong của lớp 2A: 1731
-------------------------------------------------
Tổng điểm các môn của học sinh  0  :  48
Tổng điểm các môn của học sinh  1  :  60
Tổng điểm các môn của học sinh  2  :  47
Tổng điểm các môn của học sinh  3  :  86
Tổng điểm các môn của học sinh  4  :  62
Tổng điểm các môn của học sinh  5  :  68
Tổng điểm các môn của học sinh  6  :  56
Tổng điểm các môn của học sinh  7  :  54
Tổng điểm các môn của học sinh  8  :  59
Tổng điểm các môn của học sinh  9  :  51
```

5.3 Mean, Median, Mode, Range

Statistics – Mean, Median, Mode and Range

EZY MATHS

Mean

Mean = $\frac{Total\ of\ all\ values}{number\ of\ values}$

$3, 3, 4, 5, 5, 8, 9, 15$

$\text{Mean} = \frac{52}{8} = 6.5$

Collect it all together and share it out evenly

Using the mean to find the total amount

$Mean \times Number\ of\ values$

Ezytown FC have scored an average of 3.8 goals per game in their last 15 matches. How many goals have they scored?

$3.8 \times 15 = 57 goals$

Median

Median = Middle value (Numbers written in order)

$3, 3, 4, 5, 5, 8, 9, 15$

$\text{Median} = 5$

Finds the middle value

Use of formula to find location of median

$Location = \frac{n+1}{2}$

The median of 45 values would be the 23rd number when written in order

$\frac{45+1}{2} = 23$

Mode

Mode = Most common value/item

$3, 3, 4, 5, 5, 8, 9, 15$

$\text{Mode} = 3\ and\ 5$

Average usually used for qualitative data

Occurrence of no mode

If every value appears equally, there is no mode

$1, 1, 3, 3, 7, 7$

Each value appears twice so there is no mode

Range

Range = Largest - Smallest

$3, 3, 4, 5, 5, 8, 9, 15$

$\text{Range} = 15 - 3 = 12$

Reveals how close/far apart the values are

Interpreting measures of spread

The Smaller the range, the closer and more 'consistent' the values are.

The Larger the range, the more varied and more 'inconsistent' the values are.

Mean

```
# a.mean(): Giá trị trung bình của mảng a
print('Điểm trung bình của cả lớp 2A:', diem_2a.mean())
print('-----------------------------------------------')
#Tính điểm trung bình của các học sinh trong lớp:
#CÁCH 1:
for i in range(0,diem_2a.shape[1]):
  print('Điểm trung bình của học sinh ', i,' : ', diem_2a[:,i].mean())
```

```
Điểm trung bình của cả lớp 2A: 5.77
-----------------------------------------------
Điểm trung bình của học sinh  0  :  4.8
Điểm trung bình của học sinh  1  :  6.0
Điểm trung bình của học sinh  2  :  4.7
```

```
#Tính điểm trung bình của các học sinh trong lớp:
#CÁCH 2:
mean_2a = diem_2a.mean(axis=0)
#axis = 0: theo hàng
#axis = 1: theo cột
for i in range(0,mean_2a.size):
  print('Điểm trung bình của học sinh ', i,' : ', mean_2a[i])
```

```
Điểm trung bình của học sinh  0  :  4.8
Điểm trung bình của học sinh  1  :  6.0
Điểm trung bình của học sinh  2  :  4.7
```

Mean

$$\text{Mean} = \frac{Total\ of\ all\ values}{number\ of\ values}$$

$$3, 3, 4, 5, 5, 8, 9, 15$$

$$\text{Mean} = \frac{52}{8} = 6.5$$

Collect it all together and share it out evenly

Using the mean to find the total amount

$$Mean \times Number\ of\ values$$

Ezytown FC have scored an average of 3.8 goals per game in their last 15 matches. How many goals have they scored?

$$3.8 \times 15 = 57 goals$$

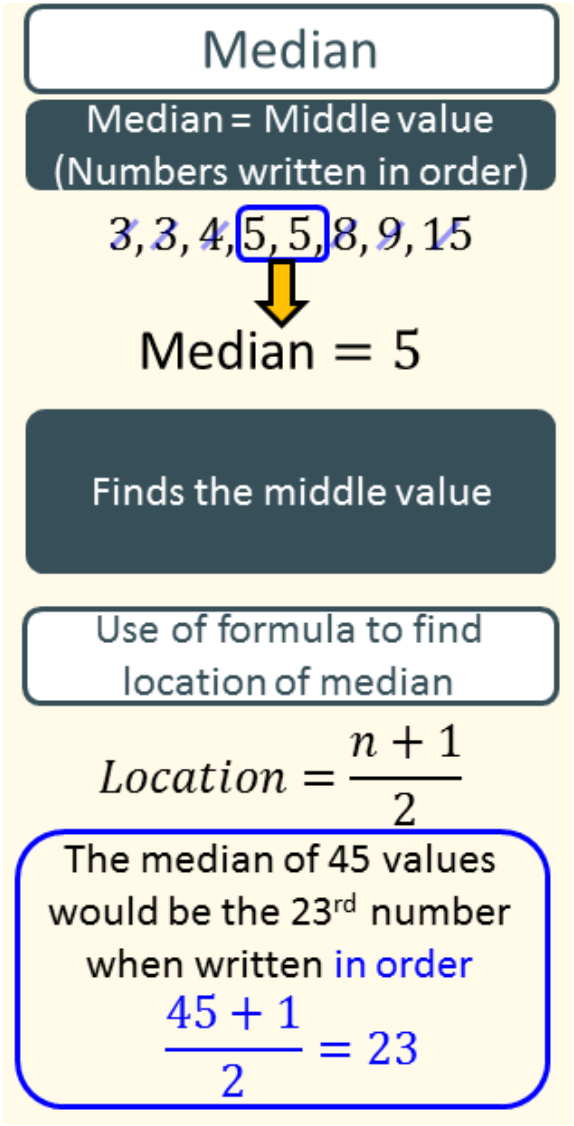
Median

- **np.median(a):Tìm trung vị của mảng a**

```
#median(): Giá trị trung vị trong một tập hợp các phần tử.
#Trường hợp số phần tử trong mảng là lẻ
a=diem_2a[1,:15]

print('Mảng a ban đầu: \n', a)
print('Số phần tử trong mảng a: ', a.size)
print('Mảng a đã sắp xếp: \n',np.sort(a,))
print('Giá trị trung bình mean:', np.mean(a))
print('Giá trị trung vị median:', np.median(a))
```

```
Mảng a ban đầu:
 [ 3  5  3 10  9  1  9  8  3  1  6  0  7 10  8]
Số phần tử trong mảng a:  15
Mảng a đã sắp xếp:
 [ 0  1  1  3  3  3  5  6  7  8  8  9  9 10 10]
Giá trị trung bình mean: 5.533333333333333
Giá trị trung vị median: 6.0
```

```
Mảng a ban đầu:
 [ 9  1  1  8  4  7  3  7  1 10]
Số phần tử trong mảng a:  10
Mảng a đã sắp xếp:
 [ 1  1  1  3  4  7  7  8  9 10]
Giá trị trung bình mean: 5.1
Giá trị trung vị median: 5.5
```



Mode

```python
#C) Mode: là giá trị xuất hiện nhiều nhất trong tập hợp.
#Trong trường hợp không có giá trị nào được lặp lại thì không có Mode.
#Liệt kê điểm xuất hiện nhiều nhất theo từng môn học
from scipy import stats as sp #sử dụng thư viện scipy để dùng hàm mode

for i in range(0,diem_2a.shape[0]):
  a = sp.mode(diem_2a[i,:])
  print('Môn ', i,': Điểm xuất hiện nhiều nhất: ', a[0],
        ' số lần: ', a[1])
print(type(a))
```

```
Môn  0 : Điểm xuất hiện nhiều nhất:  [1]  số lần:  [6]
Môn  1 : Điểm xuất hiện nhiều nhất:  [1]  số lần:  [6]
Môn  2 : Điểm xuất hiện nhiều nhất:  [9]  số lần:  [8]
Môn  3 : Điểm xuất hiện nhiều nhất:  [6]  số lần:  [5]
Môn  4 : Điểm xuất hiện nhiều nhất:  [4]  số lần:  [6]
Môn  5 : Điểm xuất hiện nhiều nhất:  [5]  số lần:  [5]
Môn  6 : Điểm xuất hiện nhiều nhất:  [9]  số lần:  [8]
Môn  7 : Điểm xuất hiện nhiều nhất:  [8]  số lần:  [15]
Môn  8 : Điểm xuất hiện nhiều nhất:  [8]  số lần:  [7]
Môn  9 : Điểm xuất hiện nhiều nhất:  [8]  số lần:  [10]
<class 'scipy.stats.stats.ModeResult'>
```

Mode

Mode = Most common value/item

$3, 3, 4, 5, 5, 8, 9, 15$

$\text{Mode} = 3 \; and \; 5$

Average usually used for qualitative data

Occurrence of no mode

If every value appears equally, there is no mode

$1, 1, 3, 3, 7, 7$

Each value appears twice so there is no mode

Range

Trong thư viện numpy không có hàm tính range, ta có thể xác định giá trị range bằng cách tính thông qua max - min

```
#D) Range: là sự khác biệt, khoảng cách giữa phần tử dưới và phần tử trên,
#giữa giá trị nhỏ nhất (Min) với giá trị lớn nhất (Max) trong tập hợp.
#Xác định độ chênh điểm max - min của từng học sinh

for i in range(0,diem_2a.shape[1]):
  print('Độ chênh điểm của học sinh ', i,' : ',
        diem_2a[:,i].max()-diem_2a[:,i].min())
```

```
Độ chênh điểm của học sinh  0  :  8
Độ chênh điểm của học sinh  1  :  7
Độ chênh điểm của học sinh  2  :  8
Độ chênh điểm của học sinh  3  :  3
Độ chênh điểm của học sinh  4  :  7
Độ chênh điểm của học sinh  5  :  9
Độ chênh điểm của học sinh  6  :  10
Độ chênh điểm của học sinh  7  :  7
Độ chênh điểm của học sinh  8  :  6
Độ chênh điểm của học sinh  9  :  9
Độ chênh điểm của học sinh  10  :  7
```

Range

Range = Largest - Smallest

$$3, 3, 4, 5, 5, 8, 9, 15$$

$$\text{Range} = 15 - 3 = 12$$

Reveals how close/far apart the values are

Interpreting measures of spread

The Smaller the range, the closer and more 'consistent' the values are.

The Larger the range, the more varied and more 'inconsistent' the values are.

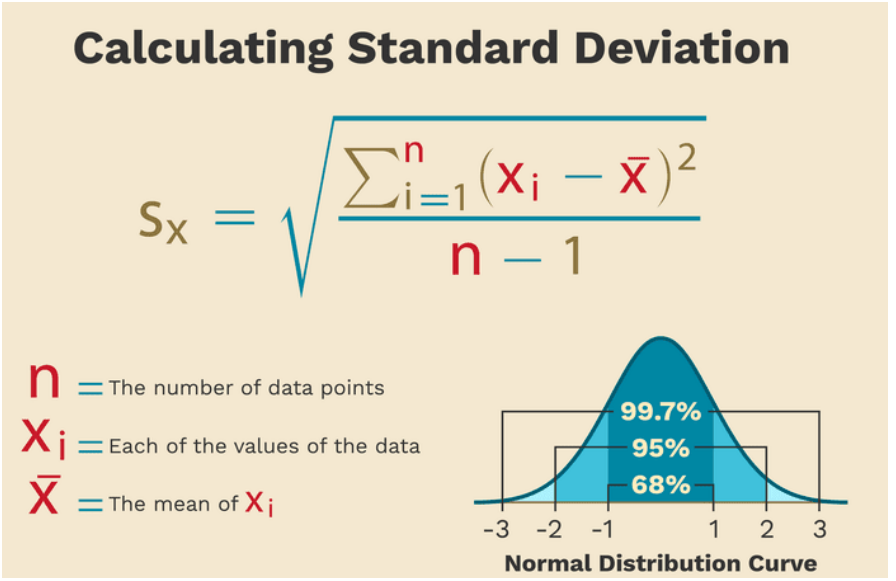
5.4 std

Độ lệch tiêu chuẩn *(standard deviation)* là đại lượng thường được sử dụng để phản ánh **mức độ phân tán** của một **biến số xung quanh số bình quân**.

```
#E) Std: Tính độ lệch chuẩn
a = np.array([10,1,1,9,12,1,9,12,10])
print('Phần tử của mảng a:',a)
print('Giá trị trung bình:',a.mean())
print('Độ lệch chuẩn:',a.std())

print('-----------------------------------------')
b = np.array([7,7,8,7,8,7,7,7,7])
print('Phần tử của mảng b:',b)
print('Giá trị trung bình:',b.mean())
print('Độ lệch chuẩn:',b.std())
```

```
Phần tử của mảng a: [10  1  1  9 12  1  9 12 10]
Giá trị trung bình: 7.222222222222222
Độ lệch chuẩn: 4.516089207311461
-----------------------------------------
Phần tử của mảng b: [7 7 8 7 8 7 7 7 7]
Giá trị trung bình: 7.222222222222222
Độ lệch chuẩn: 0.41573970964154905
```

- Nếu độ lệch chuẩn bằng 0, suy ra các giá trị quan sát cũng chính là giá trị trung bình. Nói cách khác là không có sự biến thiên.
- Nếu độ lệch chuẩn càng lớn, suy ra sự biến thiên xung quanh giá trị trung bình càng lớn.



THỰC HÀNH 2

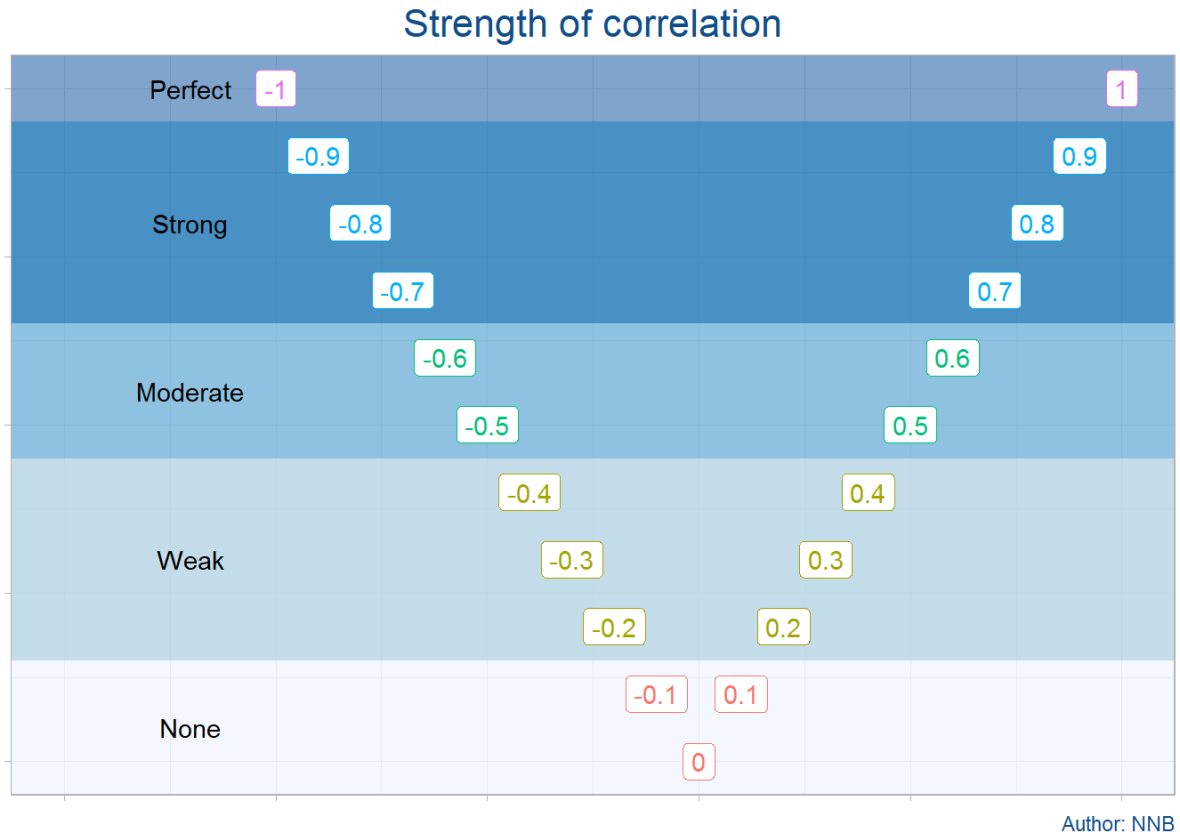
5.5 Hệ số tương quan

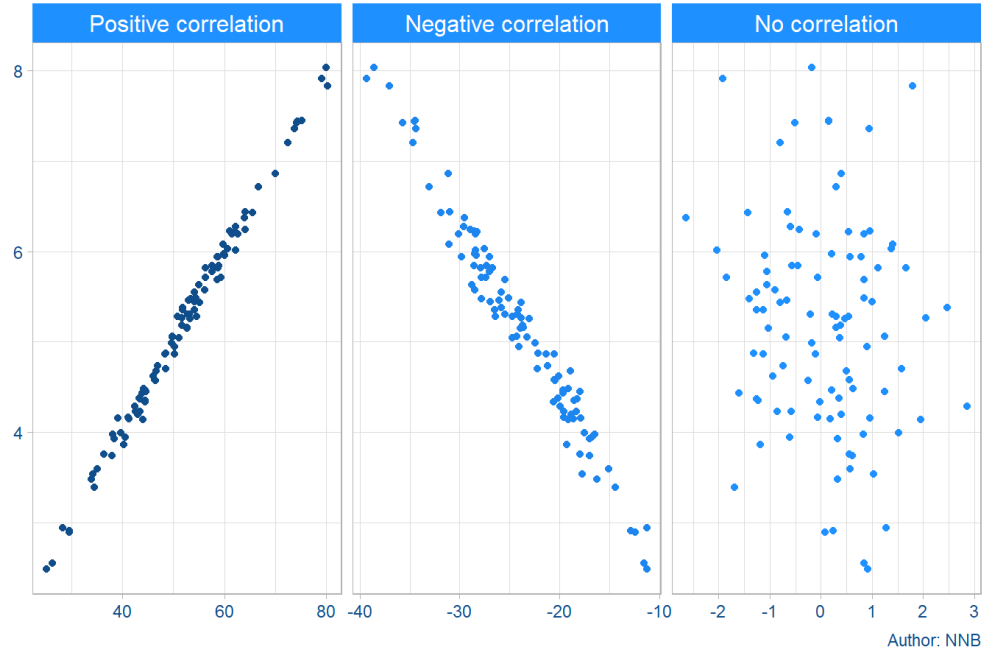
Hệ số tương quan đo lường mức độ quan hệ tuyến tính giữa hai biến.

- Hệ số tương quan không có đơn vị
- Hệ số tương quan nằm trong khoảng [-1,1]

Correlation Coefficient Formula

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$



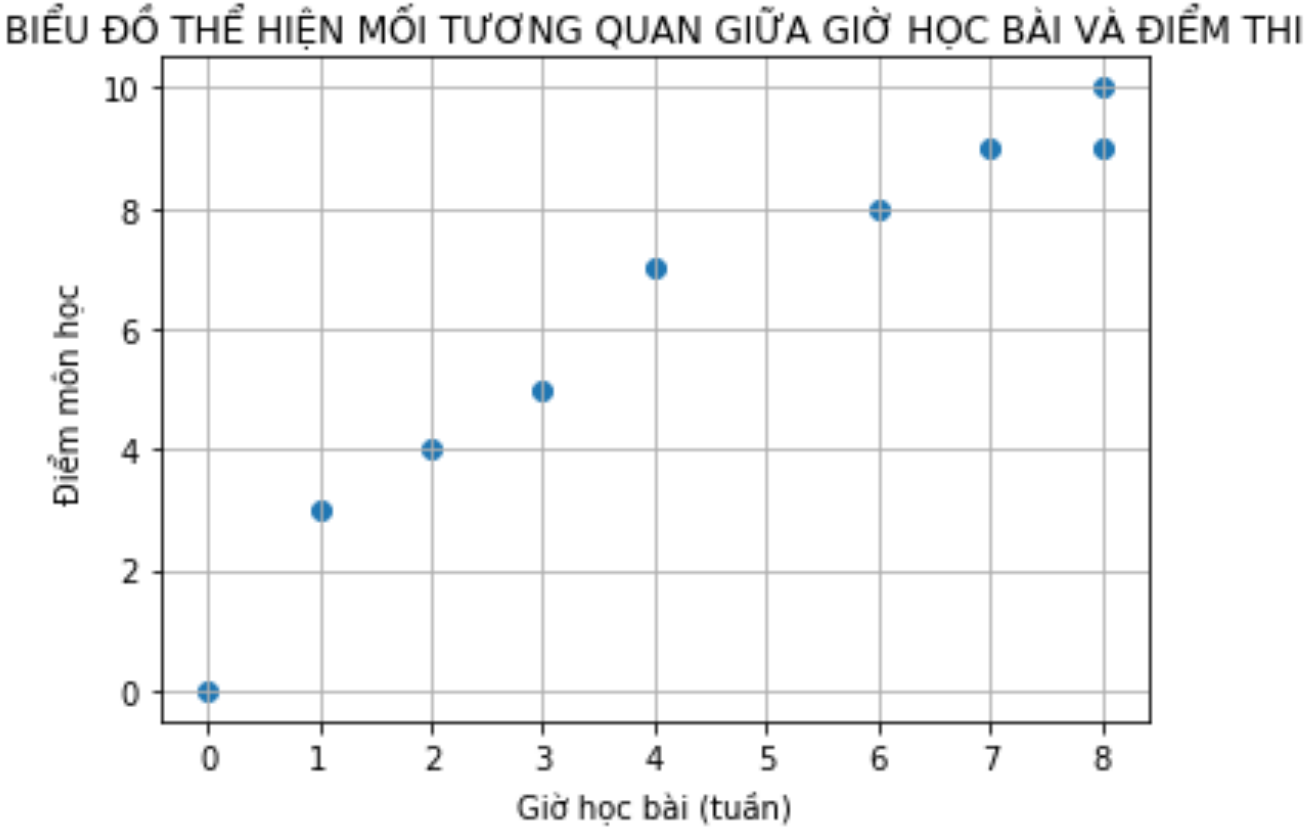


Hệ số tương quan (2)

```
#corrcoef: Hệ số tương quan
#Thời gian dành cho học bài
a_giohoc = np.array([4,7,1,2,8,0,3,8,6])
#Điểm thi nhận được:
b_diem   = np.array([7,9,3,4,9,0,5,10,8])
co = np.corrcoef(a_giohoc,b_diem)
print(type(co))
print('Hệ số tương quan: \n', co)
```

```
<class 'numpy.ndarray'>
Hệ số tương quan:
 [[1.         0.96995403]
 [0.96995403 1.        ]]
```

Ví dụ về mối tương quan giữa **thời gian dành cho việc học bài** với **điểm thi nhận được**!



THỰC HÀNH 3,4,5

Q & A
Thank you!