



Bài giảng môn học:

Khoa Học Dữ Liệu (7080509)

CHƯƠNG 4: MỘT SỐ THƯ VIỆN PYTHON TRONG KHOA HỌC DỮ LIỆU (Phần 01)

Nội dung chương 4



4.1 Giới thiệu một số thư viện Python trong KHDL

4.2 Thư viện Numpy *

4.3 Thư viện Pandas *

4.4 Thư viện Matplotlib

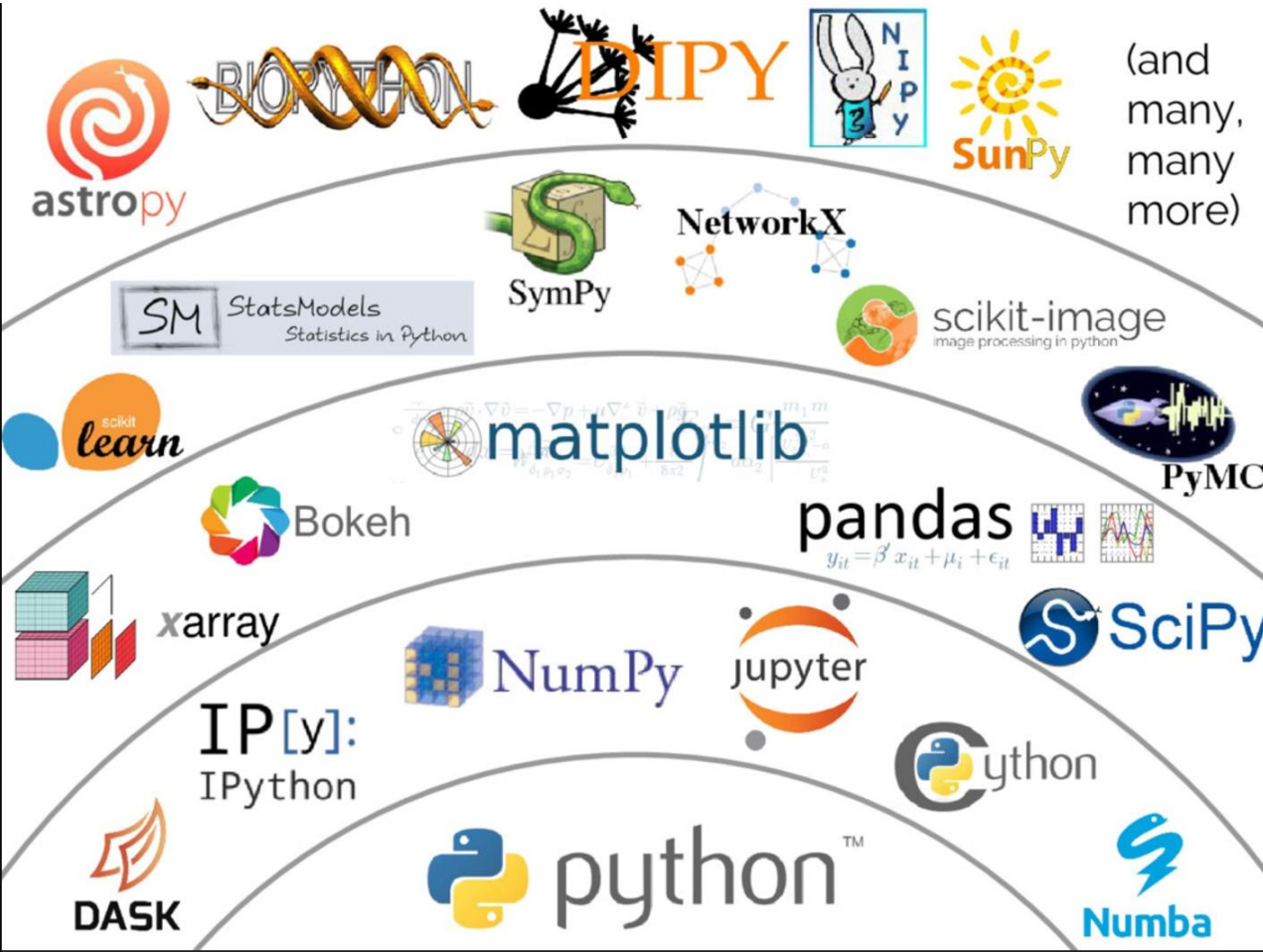
4.5 Thư viện Scikit-learn



1. Một số thư viện sử dụng trong KHDL

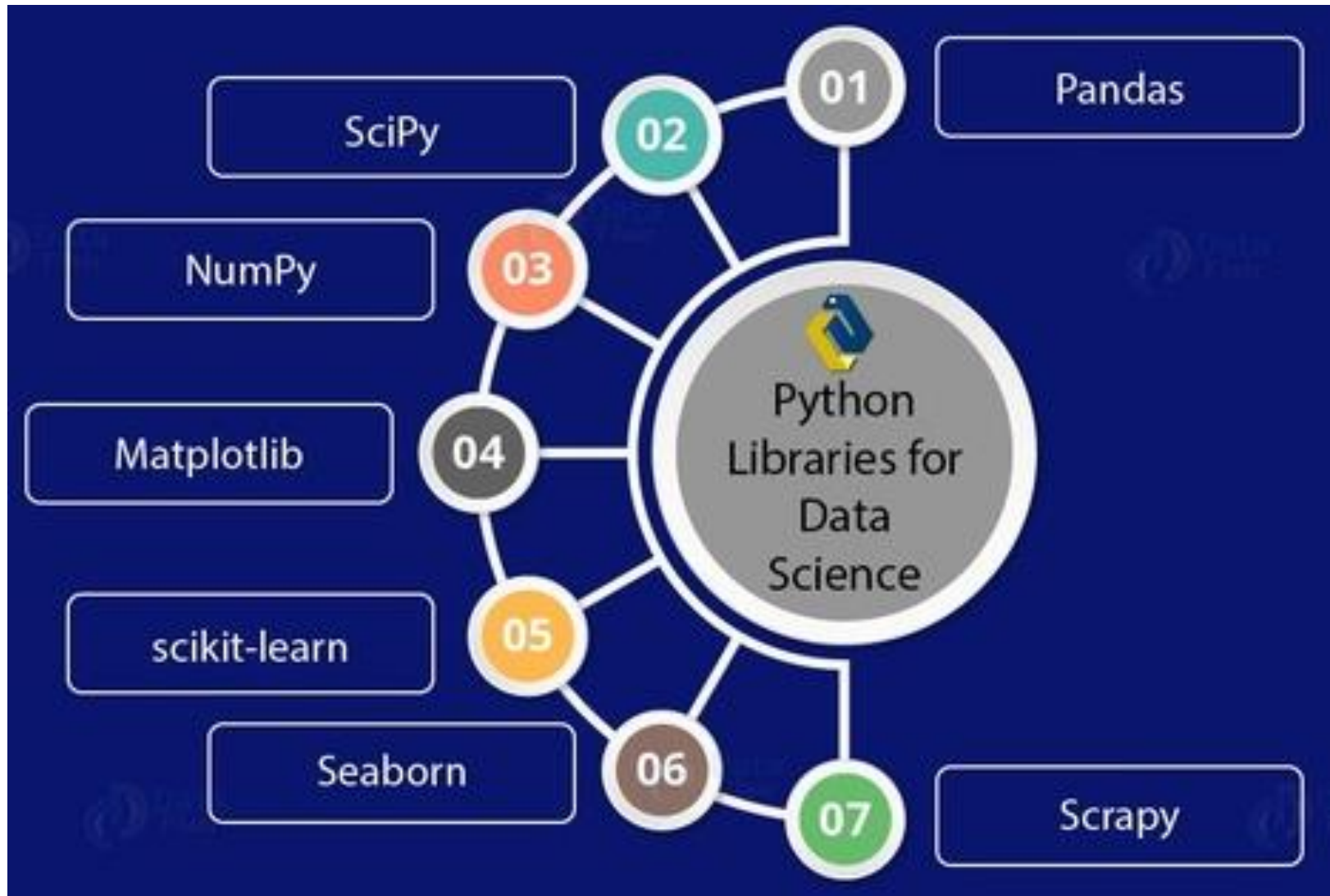


- **Python** có hệ thống thư viện rất phong phú, hỗ trợ nhiều lĩnh vực khác nhau.



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

1. Một số thư viện sử dụng trong KHDL



1. Một số thư viện sử dụng trong KHDL



01

Pandas: Sử dụng cho quản lý và tương tác với dữ liệu có cấu trúc, được sử dụng rộng rãi trong việc thu thập và tiền xử lý dữ liệu.

02

SciPy: Dựa trên Numpy, cung cấp các công cụ mạnh cho khoa học và kỹ nghệ, như biến đổi fourier rời rạc, đại số tuyến tính, tối ưu hóa và ma trận thưa

03

Numpy: Thư viện chuyên xử lý dữ liệu số (nhiều chiều), bao gồm cả các hàm đại số tuyến tính cơ bản, biến đổi fourier, sinh số ngẫu nhiên nâng cao,...

04

Matplotlib: Thư viện này được sử dụng để trực quan hóa dữ liệu (Data Visualization), chuyên vẽ các biểu đồ, hỗ trợ rất nhiều loại biểu đồ khác nhau...

1. Một số thư viện sử dụng trong KHDL



05

Scikit-learn: Thư viện chuyên về học máy; thư viện này có sẵn nhiều công cụ hiệu quả cho học máy và thiết lập các mô hình thống kê như các thuật toán phân lớp, hồi quy, phân cụm và giảm chiều dữ liệu...

06

Seaborn: Thư viện này dựa trên Matplotlib, cung cấp các công cụ hiển thị dữ liệu một cách trực quan, hiệu quả. Mục tiêu của thư viện này là sử dụng việc trực quan hóa dữ liệu như là trọng tâm của khám phá và hiểu dữ liệu

07

Scrapy: Thư viện này chuyên về việc thu thập thông tin trên Web, rất phù hợp với việc lấy các dữ liệu theo mẫu.

1. Một số thư viện sử dụng trong KHDL



Cài đặt các thư viện Python

Một số thư viện Python được cài đặt mặc định, để kiểm tra thư viện đã được cài đặt hay chưa và phiên bản đang sử dụng là bao nhiêu:

A.Sử dụng Jupyter notebook

```
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__)
```

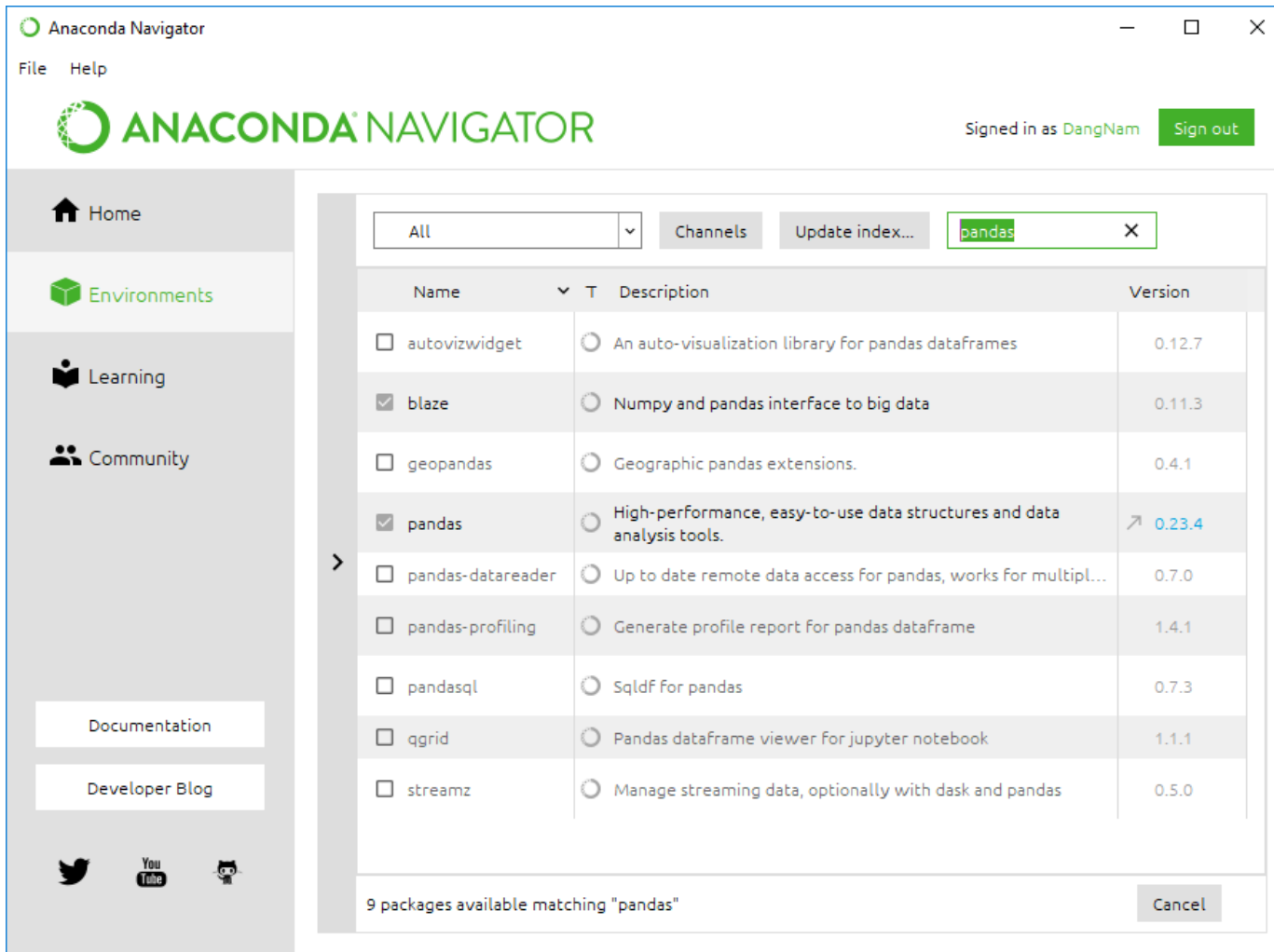
Thu vien Numpy, Version: 1.15.4

```
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'
```


Cài đặt các thư viện Python

B.Sử dụng Anaconda Navigator



The screenshot shows the Anaconda Navigator application window. The left sidebar contains navigation links: Home, Environments, Learning, and Community. The main panel displays a search results table for the 'pandas' package. The table has columns for Name, Description, and Version. The 'pandas' package is highlighted with a checkmark in the selection column. Below the table, it indicates that 9 packages are available matching 'pandas'.

Name	Description	Version
<input type="checkbox"/> autovizwidget	An auto-visualization library for pandas dataframes	0.12.7
<input checked="" type="checkbox"/> blaze	Numpy and pandas interface to big data	0.11.3
<input type="checkbox"/> geopandas	Geographic pandas extensions.	0.4.1
<input checked="" type="checkbox"/> pandas	High-performance, easy-to-use data structures and data analysis tools.	0.23.4
<input type="checkbox"/> pandas-datareader	Up to date remote data access for pandas, works for multipl...	0.7.0
<input type="checkbox"/> pandas-profiling	Generate profile report for pandas dataframe	1.4.1
<input type="checkbox"/> pandasql	SqlDF for pandas	0.7.3
<input type="checkbox"/> qgrid	Pandas dataframe viewer for jupyter notebook	1.1.1
<input type="checkbox"/> streamz	Manage streaming data, optionally with dask and pandas	0.5.0

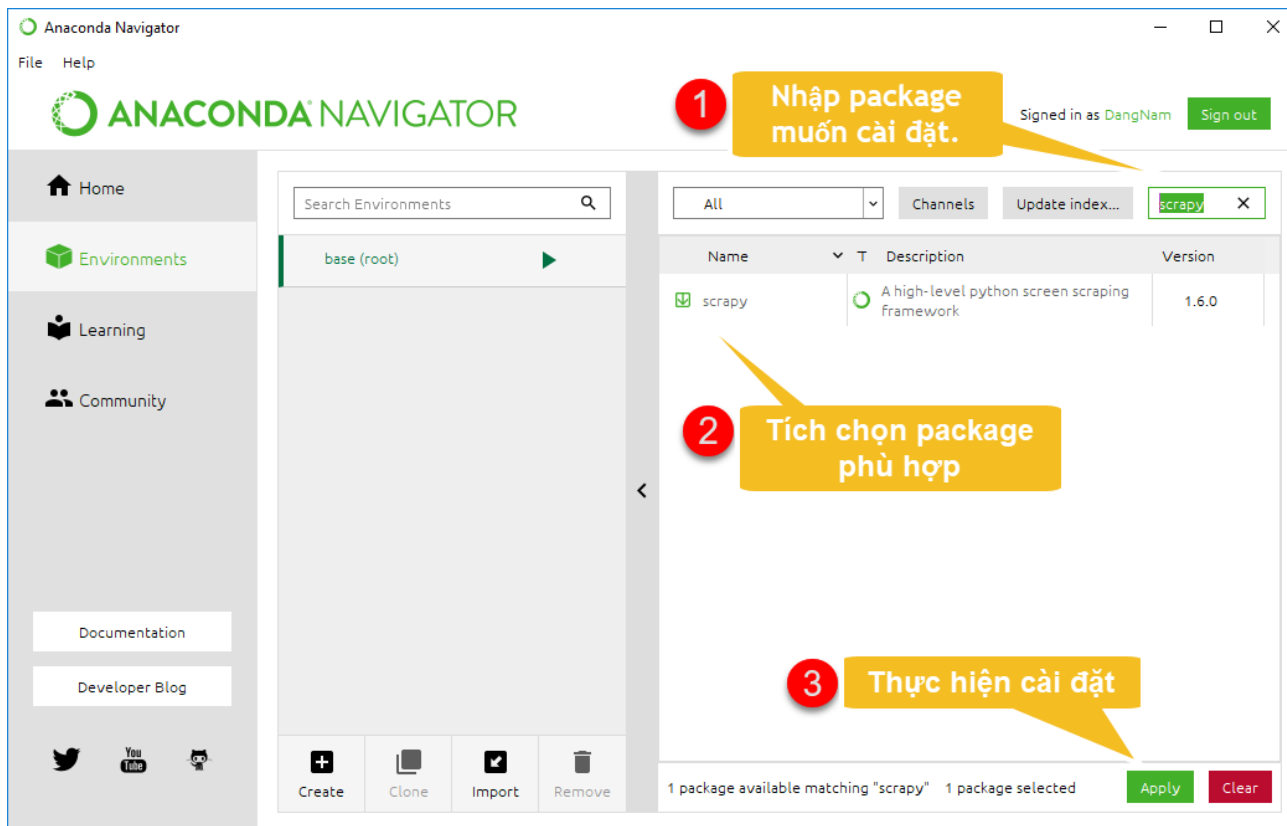
9 packages available matching "pandas"

Cài đặt các thư viện Python

Trường hợp thư viện chưa được cài đặt, có thể sử dụng lệnh:

!pip install <tên thư viện>

Hoặc sử dụng Anaconda Navigator:



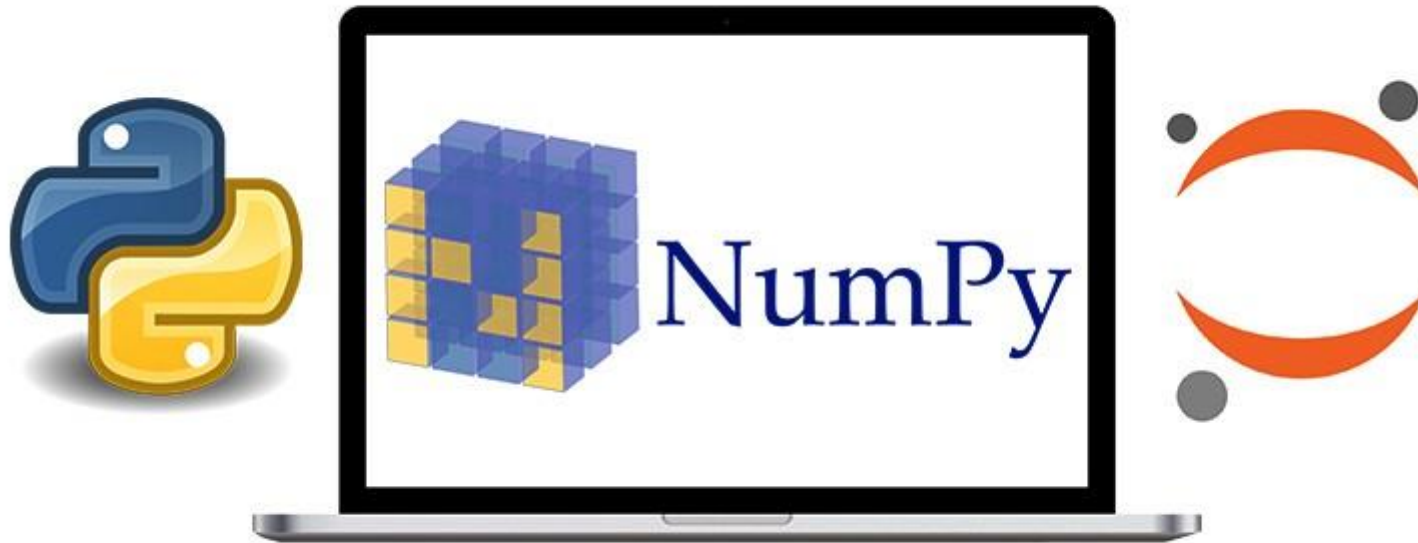
The screenshot shows the Anaconda Navigator application window. The left sidebar contains navigation options: Home, Environments, Learning, and Community. The main area displays the 'base (root)' environment. A search bar at the top right of the main area is used to find packages. A table of available packages is shown, with 'scrapy' selected. The status bar at the bottom indicates '1 package available matching "scrapy" 1 package selected' and provides 'Apply' and 'Clear' buttons.

1 Nhập package muốn cài đặt.

2 Tích chọn package phù hợp

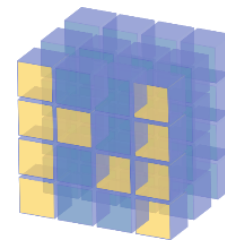
3 Thực hiện cài đặt

2. Thư viện NumPy



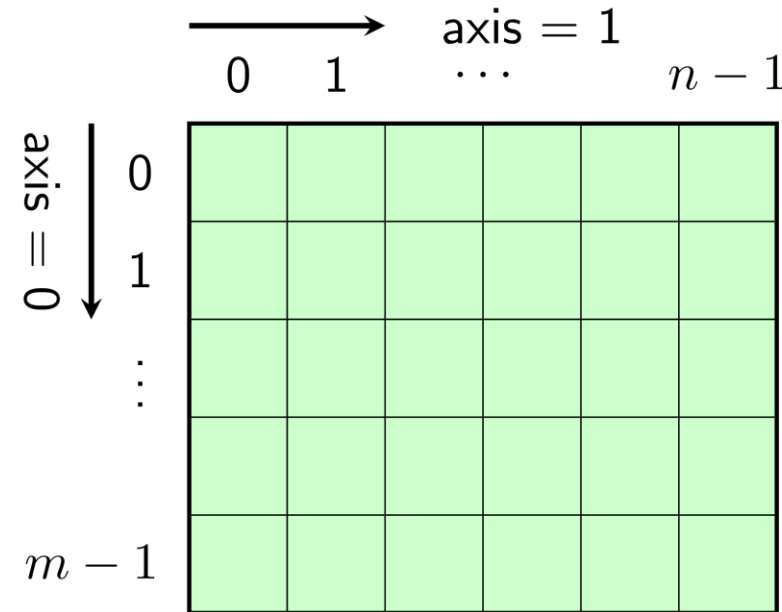
2. Thư viện Numpy

- **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/>



2. 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 ma trận



2. Thư viện Numpy

2.1) Khởi tạo mảng

1	5	18	23
---	---	----	----

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

3	12	66
7	9	34
23	45	11

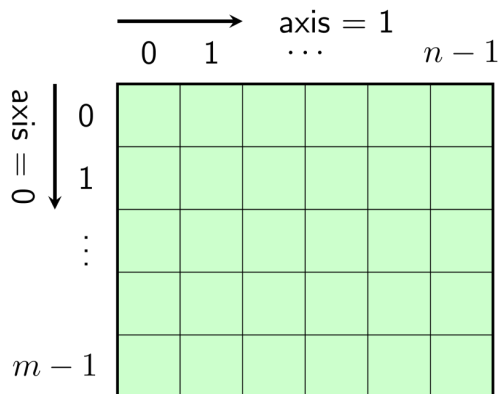
Matrix (2D array)
Dimension = 2
(2 indexes required)

3	12	66
7	9	34
23	45	11

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

3	12	66
7	9	34
23	45	11

ND array
Dimension = N
(N indexes required)



2.1 Khởi tạo mảng

- Khởi tạo mảng 1 chiều – 1D (Vector)

```
1 #Khởi tạo mảng 1 chiều với thư viện Numpy
2 import numpy as np
3
4 #Tạo mảng 1 chiều (1D)
5 a = np.array((1, 2, 5, 7, 0, 8))
6
7 print(a)
8 print("Loại dữ liệu của biến a:", type(a))
9 print("Kiểu dữ liệu của phần tử trong mảng a:", a.dtype)
10 print("Kích thước của mảng a:", a.shape)
11 print("Số phần tử của mảng a:", a.size)
12 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

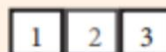
Kích thước của mảng a: (6,)

Số phần tử của mảng a: 6

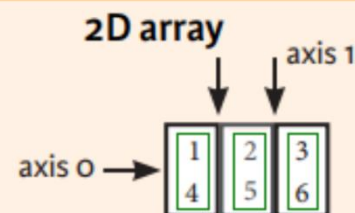
Số chiều của mảng a: 1

NumPy Arrays

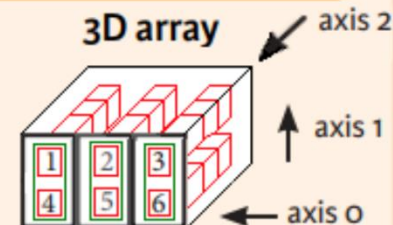
1D array



2D array



3D array



2.1 Khởi tạo mảng

- Chuyển đổi dữ liệu từ list sang – mảng 1D (Vector)

```
1 import numpy as np
2 #Chuyển đổi từ biến kiểu list sang biến mảng
3 list_a = [8, 6, 5, 7.2, 12, 1]
4 print("Danh sách list_a:", list_a)
5
6 #chuyển sang kiểu array
7 array_a = np.array(list_a)
8
9 print("Mảng array_a:", array_a)
10 print("Loại dữ liệu của biến array_a:", type(array_a))
11 print("Kiểu dữ liệu của phần tử trong mảng array_a:", array_a.dtype)
12 print("Kích thước của mảng array_a:", array_a.shape)
13 print("Số phần tử của mảng array_a:", array_a.size)
14 print("Số chiều của mảng array_a:", array_a.ndim)
```

Danh sách list_a: [8, 6, 5, 7.2, 12, 1]

Mảng array_a: [8. 6. 5. 7.2 12. 1.]

Loại dữ liệu của biến array_a: <class 'numpy.ndarray'>

Kiểu dữ liệu của phần tử trong mảng array_a: float64

Kích thước của mảng array_a: (6,)

Số phần tử của mảng array_a: 6

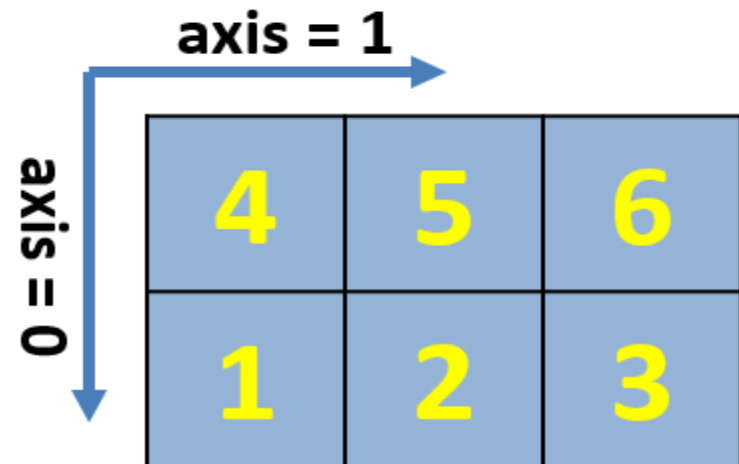
Số chiều của mảng array_a: 1

2.1 Khởi tạo mảng

- Khởi tạo mảng 2 chiều – 2D (Matrix)

```
1 #Gọi thư viện numpy
2 import numpy as np
3
4 #Tạo mảng 2 chiều (2D - Ma trận)
5 b = np.array([(4, 5, 6.0),(1, 2, 3.5)])
6
7 print(b)
8 print("Loại dữ liệu của biến b:", type(b))
9 print("Kiểu dữ liệu của phần tử trong mảng b:", b.dtype)
10 print("Kích thước của mảng b:", b.shape)
11 print("Số phần tử của mảng b:", b.size)
12 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
```



2.1 Khởi tạo mảng

• Khởi tạo mảng 3 chiều – 3D

```

1 import numpy as np
2
3 c = np.array([[ (2,4,0,6), (4,7,5,6)],
4               [ (0,3,2,1), (9,4,5,6)],
5               [ (5,8,6,4), (1,4,6,8)]] ) #mảng 3 chiều (3D)
6
7 print(c)
8 print("Phần tử đầu tiên của mảng c:",c[0,0,0])
9 print("Kiểu dữ liệu của phần tử trong mảng c:",c.dtype)
10 print("Kích thước của mảng c:",c.shape)
11 print("Số phần tử của mảng c:",c.size)
12 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]]]

```

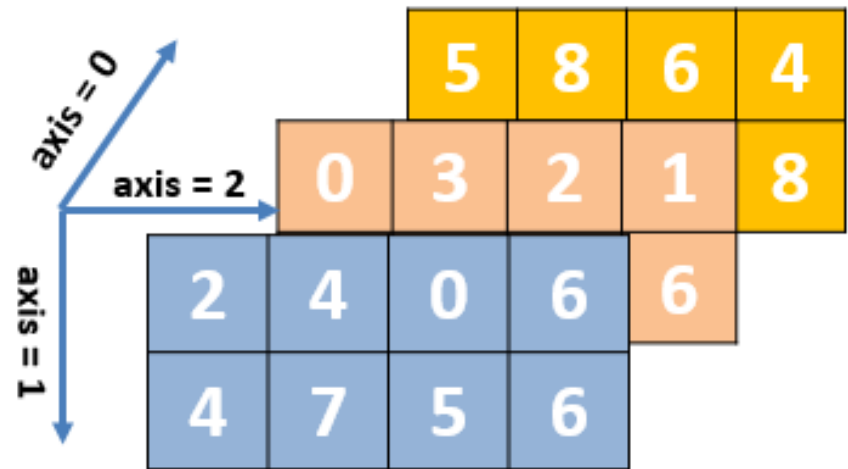
Phần tử đầu tiên của mảng c: 2

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



2.1 Khởi tạo mảng

- Khởi tạo mảng với các hàm sẵn có của Numpy

Initial Placeholders

```
>>> np.zeros((3,4))
>>> np.ones((2,3,4),dtype=np.int16)
>>> d = np.arange(10,25,5)

>>> np.linspace(0,2,9)

>>> e = np.full((2,2),7)
>>> f = np.eye(2)
>>> np.random.random((2,2))
>>> np.empty((3,2))
```

Create an array of zeros
Create an array of ones
Create an array of evenly spaced values (step value)
Create an array of evenly spaced values (number of samples)
Create a constant array
Create a 2X2 identity matrix
Create an array with random values
Create an empty array

2. Thư viện Numpy

- Ví dụ:

```
1 # Tạo ma trận 0 kích thước 5 hàng x 3 cột
2 import numpy as np
3
4 array_zeros = np.zeros((5, 3))
5
6 print(array_zeros)
7 print("Kiểu dữ liệu trong mảng array_zeros:", array_zeros.dtype)
8 print("Kích thước của mảng array_zeros:", array_zeros.shape)
9 print("Số phần tử của mảng array_zeros:", array_zeros.size)
10 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 của phần tử 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

2. Thư viện Numpy

- Ví dụ:

```
1 #Tạo ma trận đơn vị vuông cấp 5
2 import numpy as np
3 array_eye = np.eye(5)
4
5 print(array_eye)
6 print("Kiểu dữ liệu của phần tử trong mảng array_eye:", array_eye.dtype)
7 print("Kích thước của mảng array_eye:", array_eye.shape)
8 print("Số phần tử của mảng array_eye:", array_eye.size)
9 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

2. Thư viện Numpy

- Ví dụ:

```
1  #Tạo một ma trận 1 chiều bao gồm 10 phần tử ngẫu nhiên [0,1]
2  import numpy as np
3  array_random = np.random.random((10))
4
5  print(array_random)
6  print("Kiểu dữ liệu của phần tử trong mảng array_random:", array_random.dtype)
7  print("Kích thước của mảng array_random:", array_random.shape)
8  print("Số phần tử của mảng array_random:", array_random.size)
9  print("Số chiều của mảng array_random:", array_random.ndim)
```

```
[0.48963841 0.72817439 0.12369405 0.64774516 0.28791091 0.71088151
 0.31917933 0.16395153 0.50415822 0.99443047]
```

Kiểu dữ liệu của phần tử trong mảng array_random: float64

Kích thước của mảng array_random: (10,)

Số phần tử của mảng array_random: 10

Số chiều của mảng array_random: 1

2.1 Khởi tạo mảng



Yêu cầu: Tạo một ma trận 10 x 10, bao gồm các phần tử, là những số nguyên ngẫu nhiên trong khoảng (0-100)

```
1  '''
2  Bài tập: Tạo một ma trận 10x10 các số
3  nguyên ngẫu nhiên nằm trong khoảng [0-100]
4  '''
5  import numpy as np
6  b=np.random.random((10,10))*100
7  #Phần tử mặc định tạo ra có kiểu float
8  #Chuyển sang kiểu số nguyên int
9  b = b.astype(np.int64)
10 print(b)
11 print("Kiểu dữ liệu của các phần tử: ", b.dtype)
12 print("Số chiều của mảng: ", b.ndim)
13 print("Kích thước của ma trận: ", b.shape)
14
```

```
[[ 7 87 84 49 96 50 14 87 64 26]
 [35 94 84 48  3 84 37 32 36 39]
 [58 47 23 11 67 58 94 73 97 77]
 [72 26 91 54 78 45 30 46 83 77]
 [56 76 61 36 50 37 24 88 47 98]
 [ 4 31 83 27 63 26 17 54 12 17]
 [ 6 79 27 47 30 64 78 15 60 41]
 [38 65 20 28 22 97  2 63 50 58]
 [77 55 67 76 20 74 29 86 82 24]
 [66 42 44 56 21 42  0 98 88 90]]
```

Kiểu dữ liệu của các phần tử: int64

Số chiều của mảng: 2

Kích thước của ma trận: (10, 10)

2.2 Quan sát mảng

- Bảng điểm của lớp 2A** (bao gồm 30 học sinh, tương ứng với 30 cột, của 10 môn học, tương ứng với 10 hàng)

Diem_2A - Notepad

Điểm học sinh i (30 học sinh)																													
2	4	3	7	5	6	5	6	8	9	3	6	1	9	8	7	3	3	9	5	1	6	5	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						
1	10	4	9	6	9	0	2	3	1	8	6	8	4	2	9	2	9	5	0	4	1	7	3						
6	3	0	8	3	7	7	2	6	8	7	3	4	1	5	9	1	0	2	10	4	6	8	6						
4	3	6	7	4	5	2	6	9	4	3	9	9	4	5	7	2	10	9	4	0	5	3	1						
2	3	8	10	4	5	9	5	4	7	10	1	8	4	3	9	6	3	6	7	4	7	3	5						
9	9	1	10	9	9	5	9	6	3	9	5	1	10	7	10	2	8	8	1	8	4	5	4						
8	8	4	8	0	4	4	8	6	7	1	3	1	6	8	8	4	6	8	4	0	1	8	2						
6	7	8	9	10	9	2	2	6	1	10	9	6	3	9	5	9	8	1	1	8	8	8	6						
7	8	7	8	6	10	10	6	8	10	8	9	8	8	5	10	8	7	8	7	9	9	8	7						

Điểm của môn học j (10 môn học)

- Đọc dữ liệu từ file Diem_2A.txt**

```
1 #Khai báo sử dụng thư viện Numpy
2 import numpy as np
3 #đọc file Diem_2A.txt
4 path = 'Data_C4\Diem_2A.txt'
5
6 diem_2a = np.loadtxt(path, delimiter=',')
7 print('File dữ liệu điểm lớp 2A:\n', diem_2a)
```

File dữ liệu điểm lớp 2A:

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

2.2 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

2.2 Quan sát mảng

```
#a.astype(kiểu mới): Chuyển đổi kiểu dữ liệu của mảng a  
#Chuyển dữ liệu của mảng diem_2a từ float sang kiểu int
```

```
diem_2a_int = diem_2a.astype(np.int16)  
print('Dữ liệu mảng diem_2a sau khi chuyển: ', diem_2a_int.dtype)
```

Dữ liệu mảng diem_2a sau khi chuyển: int16

```
#Chuyển dữ liệu sang kiểu string:  
diem_2a_str = diem_2a.astype(np.str)  
print('Dữ liệu mảng diem_2a sau khi chuyển: ', diem_2a_str.dtype)  
print(diem_2a_str)
```

Dữ liệu mảng diem_2a sau khi chuyển: <U32

```
[['2.0' '4.0' '3.0' '7.0' '5.0' '6.0' '5.0' '6.0' '8.0' '9.0' '3.0' '6.0'  
  '1.0' '9.0' '8.0' '7.0' '3.0' '3.0' '9.0' '5.0' '1.0' '6.0' '5.0' '1.0']
```

2.2 Quan sát mảng



NumPy dtypes

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

2.3 Các phép toán trên mảng

- Cho phép thực hiện +, -, *, /, ... các phần tử trên mảng như các phần tử riêng biệt.

```
1  #Thực hiện phép toán giữa biến array và một giá trị
2  a=np.array((7,5,8,1))
3  print('Mảng a: ', a)
4
5  x = 3
6  # Các phép toán giữa array với một giá trị
7  array_tong = a + x
8  print("Tổng mảng a + x:", array_tong)
9
10 array_hieu = a - x
11 print("Tổng mảng a - x:", array_hieu)
12
13 array_tich=a*x
14 print("Tích mảng a * b:", array_tich)
15
16 array_thuong=a/x
17 print("Tích mảng a / b:", array_thuong)
18
19 array_thuongnguyen=a//x
20 print("Thương nguyên mảng a // b:", array_thuongnguyen)
21
22 array_thuongdu=a%x
23 print("Thương dư của mảng a % b:", array_thuongdu)
```


2.3 Các phép toán trên mảng (t)

- Nếu thực hiện với 2 biến kiểu array, yêu cầu số phần tử trong 2 biến phải bằng nhau.

```
1  #Thực hiện các phép toán giữa 2 biến array
2  # Yêu cầu: Số phần tử của hai ma trận phải bằng nhau
3  a=np.array((7,5,8,1))
4  print('Mảng a: ', a)
5  b=np.array((1,2,3,4))
6  print('Mảng b: ', b)
7
8  #Các phép toán trên 2 biến array
9  array_tong = a + b
10 print("Tổng mảng a + b:", array_tong)
11
12 array_hieu = a - b
13 print("Tổng mảng a - b:", array_hieu)
14
15 array_tich=a*b
16 print("Tích mảng a * b:", array_tich)
17
18 array_thuong=a/b
19 print("Tích mảng a / b:", array_thuong)
20
21 array_thuongnguyen=a//b
22 print("Thương nguyên mảng a // b:", array_thuongnguyen)
23
24 array_thuongdu=a%b
25 print("Thương dư của mảng a % b:", array_thuongdu)
```

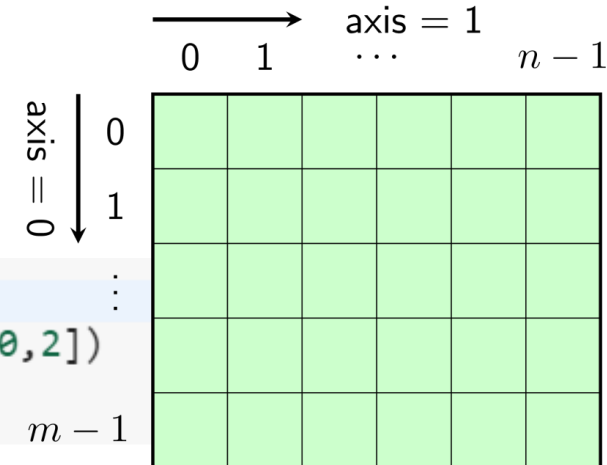
2.4 Truy cập phần tử mảng

- Truy cập tới phần tử cụ thể trong mảng sử dụng chỉ số mảng (chỉ số bắt đầu từ 0):
 - $a[i]$: truy cập tới phần tử thứ i của mảng một chiều
 - $a[i,j]$: Truy cập tới phần tử hàng i , cột j của mảng 2 chiều
 - $a[n,i,j]$: Truy cập tới phần tử chiều n , hàng i , cột j của mảng 3 chiều

```
#Truy cập phần tử của mảng:
print('các phần tử của mảng 1 chiều a:\n', a)
print('phần tử thứ 5 của mảng:', a[4]).
```

các phần tử của mảng 1 chiều a:
 [3. 5. 3. 10. **9.** 1. 9. 8. 3. 1.]
 phần tử thứ 5 của mảng: 9.0

```
#Truy cập phần tử của mảng 2 chiều:
print('Điểm môn học thứ 1, của học sinh thứ 3 là:', diem_2a[0,2])
print('Bảng điểm lớp 2A:\n', diem_2a)
```



Điểm môn học thứ 1, của học sinh thứ 3 là: 3.0
 Bảng điểm lớp 2A:

```
[[ 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.]
```

2.4 Truy cập phần tử mảng (t)

- Truy cập nhiều phần tử trong mảng sử dụng ký hiệu “.”
 - a[3:8]: Truy cập tới các phần tử thứ 4 tới phần tử thứ 9 của mảng một chiều a
 - b[:3, :] : Truy cập tới phần tử từ hàng 0 tới hàng 3, của tất cả các cột trong mảng 2 chiều

```
#Truy cập tới nhi  
#Lấy điểm tất cả  
diem_hs5 = diem_2
```

```
print("Điểm các m
```

Điểm các môn của



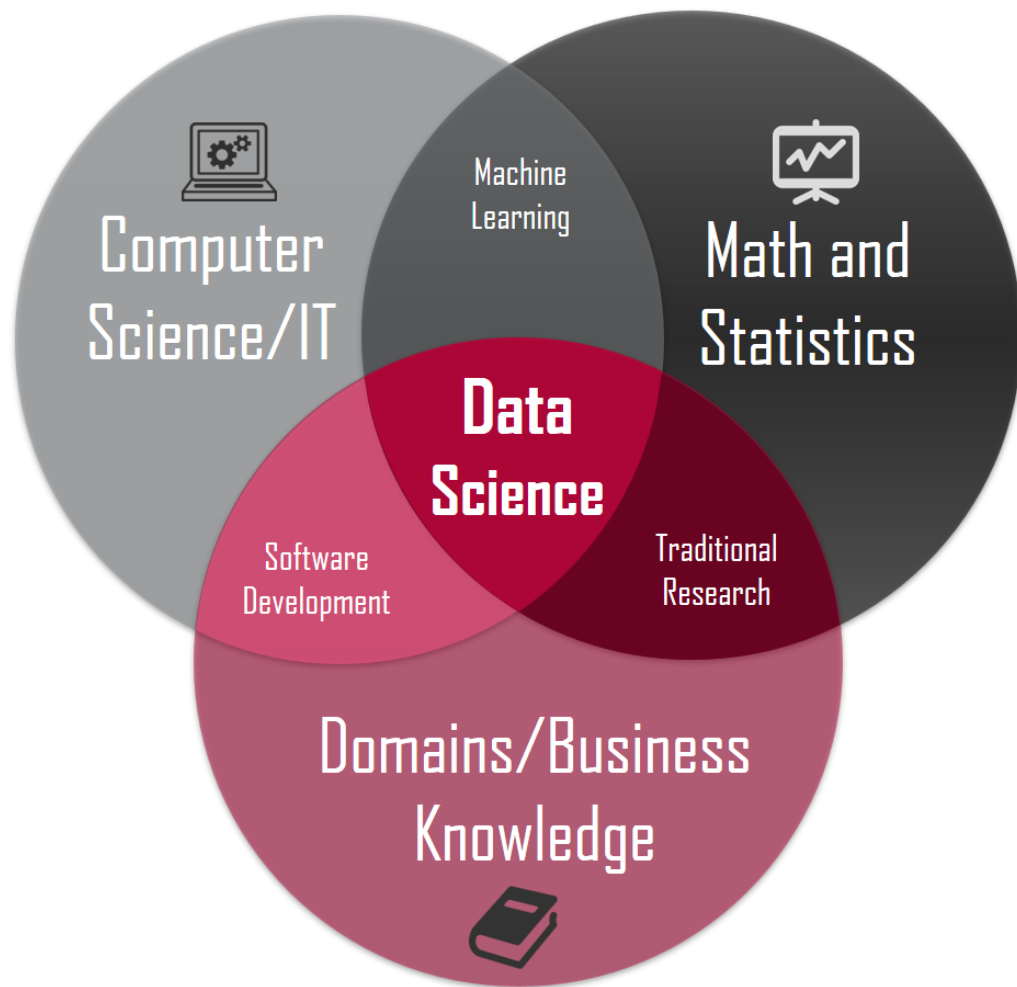
```
diem_hs = diem_2a[:,10:20]  
print("Bảng điểm từ học sinh 10 tới học sinh 20:\n",diem_hs)
```



Bảng điểm từ học sinh 10 tới học sinh 20:

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

2.5 Các hàm thống kê



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

2.5 Các hàm thống kê: Max - Min

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



```
#Xác định giá trị lớn nhất, nhỏ nhất:  
#Liệt kê điểm cao nhất và thấp nhất của mỗi môn học  
for i in range(0,diem_2a.shape[0]):  
    print('Môn ', i+1, ': Điểm Max: ', diem_2a[i,:].max(),  
          '-- Điểm Min:', diem_2a[i,:].min())
```



```
Môn 1 : Điểm Max: 9.0 -- Điểm Min: 1.0  
Môn 2 : Điểm Max: 10.0 -- Điểm Min: 0.0  
Môn 3 : Điểm Max: 10.0 -- Điểm Min: 0.0  
Môn 4 : Điểm Max: 10.0 -- Điểm Min: 0.0  
Môn 5 : Điểm Max: 10.0 -- Điểm Min: 0.0  
Môn 6 : Điểm Max: 10.0 -- Điểm Min: 1.0  
Môn 7 : Điểm Max: 10.0 -- Điểm Min: 1.0  
Môn 8 : Điểm Max: 9.0 -- Điểm Min: 0.0  
Môn 9 : Điểm Max: 10.0 -- Điểm Min: 1.0  
Môn 10 : Điểm Max: 10.0 -- Điểm Min: 5.0
```



2.5 Các hàm thống kê: Sum

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

```
#Sum:Tính tổng các phần tử
```

```
print('Tổng tất các điểm trong của lớp 2A:', np.sum(diem_2a))
```

```
#Tính tổng điểm của từng học sinh
```

```
for i in range(0,30):  
    print('Tổng điểm các môn của học sinh ', i+1, ' : ', diem_2a[:,i].sum())
```

```
Tổng tất các điểm trong của lớp 2A: 1662.0  
Tổng điểm các môn của học sinh 1 : 48.0  
Tổng điểm các môn của học sinh 2 : 60.0  
Tổng điểm các môn của học sinh 3 : 44.0  
Tổng điểm các môn của học sinh 4 : 86.0  
Tổng điểm các môn của học sinh 5 : 56.0
```



2.5 Các hàm thống kê

Statistics – Mean, Median, Mode and Range

EZY MATHS

Mean

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

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

$\text{Mean} \times \text{Number of values}$

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

$$3.8 \times 15 = 57 \text{ goals}$$

Median

Median = Middle value
(Numbers written in order)

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

Median = 5

Finds the middle value

Use of formula to find location of median

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

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

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

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.

2.5 Các hàm thống kê: Mean

```
#mean(): Giá trị trung bình
#Tính điểm trung bình của các học sinh trong lớp:
#CÁCH 1:
for i in range(0,30):
    print('Điểm trung bình của học sinh ', i+1, ' : ', diem_2a[:,i].mean())
```

```
Điểm trung bình của học sinh 1 : 4.8
Điểm trung bình của học sinh 2 : 6.0
Điểm trung bình của học sinh 3 : 4.4
Điểm trung bình của học sinh 4 : 8.6
Điểm trung bình của học sinh 5 : 5.6
```

```
#CÁCH 2:
mean_2a = diem_2a.mean(axis=0)

for i in range(0,mean_2a.size):
    print('Điểm trung bình của học sinh ', i+1, ' : ', mean_2a[i])
```

```
Điểm trung bình của học sinh 1 : 4.8
Điểm trung bình của học sinh 2 : 6.0
Điểm trung bình của học sinh 3 : 4.4
Điểm trung bình của học sinh 4 : 8.6
Điểm trung bình của học sinh 5 : 5.6
```

Mean

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

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

$\text{Mean} \times \text{Number of values}$

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

$$3.8 \times 15 = 57 \text{ goals}$$

2.5 Các hàm thống kê: Median

```
#median(): Giá trị giữa trong một tập hợp các phần tử.
#Trường hợp số phần tử trong mảng là lẻ
a=diem_2a[:,5]
b=np.array([9,8,6])
a=np.append(a,b) #nối mảng b vào mảng a

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:

[6. 1. 9. 7. 5. 5. 9. 4. 9. 10. 9. 8. 6.]

Số phần tử trong mảng a: 13

Mảng a đã sắp xếp:

[1. 4. 5. 5. 6. 6. 7. 8. 9. 9. 9. 9. 10.]

Giá trị trung bình mean: 6.769230769230769

Giá trị trung vị median: 7.0

Median

Median = Middle value
(Numbers written in order)

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



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

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

2.5 Các hàm thống kê: 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**.

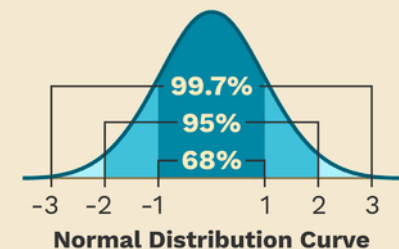
```
#Std: độ lệch chuẩn
#Tính độ lệch chuẩn điểm thi của từng học sinh
for i in range(0,30):
    print('Độ lệch chuẩn của học sinh ', i+1, ' : ', diem_2a[:,i].std())
```

```
Độ lệch chuẩn của học sinh 1 : 2.6381811916545836
Độ lệch chuẩn của học sinh 2 : 2.569046515733026
Độ lệch chuẩn của học sinh 3 : 2.65329983228432
Độ lệch chuẩn của học sinh 4 : 1.1135528
Độ lệch chuẩn của học sinh 5 : 2.9393876
Độ lệch chuẩn của học sinh 6 : 2.6925824
```

Calculating Standard Deviation

$$S_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

n = The number of data points
 x_i = Each of the values of the data
 \bar{x} = The mean of **x_i**



2.5 Các hàm thống kê: Mode

```
#Mode: phần tử xuất hiện nhiều nhất
#Liệt kê điểm xuất hiện nhiều nhất theo 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+1,': Điểm xuất hiện nhiều nhất: ', a[0],
          ' số lần: ', a[1])

print(type(a))
```

Mode

Mode = Most common
value/item

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

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

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

2.5 Các hàm thống kê: 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

#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(0,30):  
    print('Độ chênh điểm của học sinh ', i+1, ' : ',  
          diem_2a[:,i].max()-diem_2a[:,i].min())
```

```
Độ chênh điểm của học sinh 1 : 8.0  
Độ chênh điểm của học sinh 2 : 7.0  
Độ chênh điểm của học sinh 3 : 8.0  
Độ chênh điểm của học sinh 4 : 3.0  
Độ chênh điểm của học sinh 5 : 10.0  
Độ chênh điểm của học sinh 6 : 9.0
```

Range

Range = Largest - Smallest

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

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.

2.5 Các hàm thống kê: corrcoeff

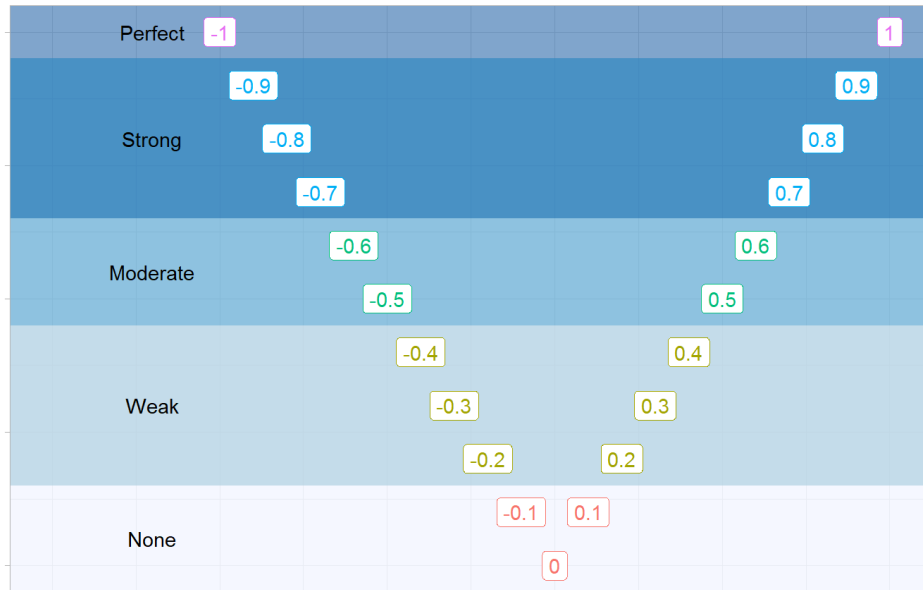
Hệ số tương quan đo lường mức độ quan hệ tuyến tính giữa hai biến.

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

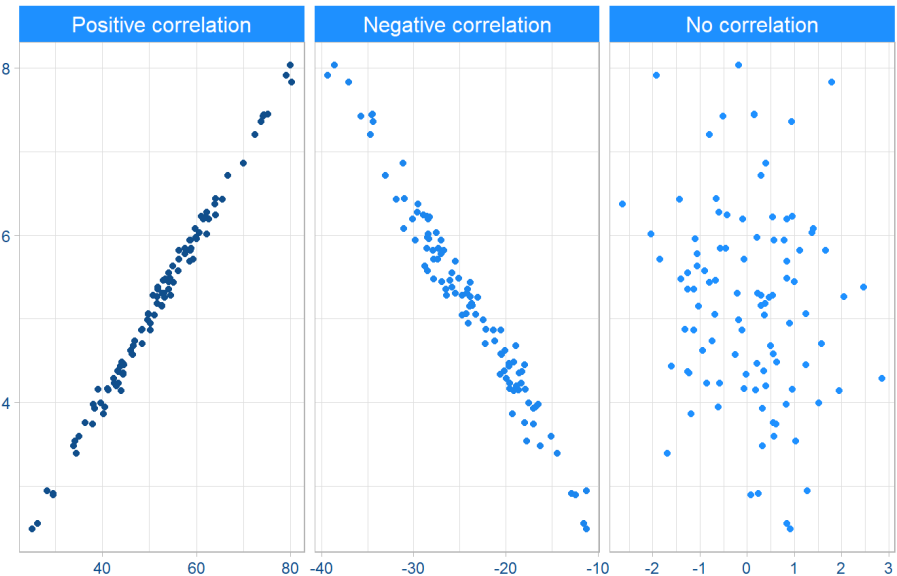
Correlation Coefficient Formula

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

Strength of correlation



Author: NNB



Author: NNB

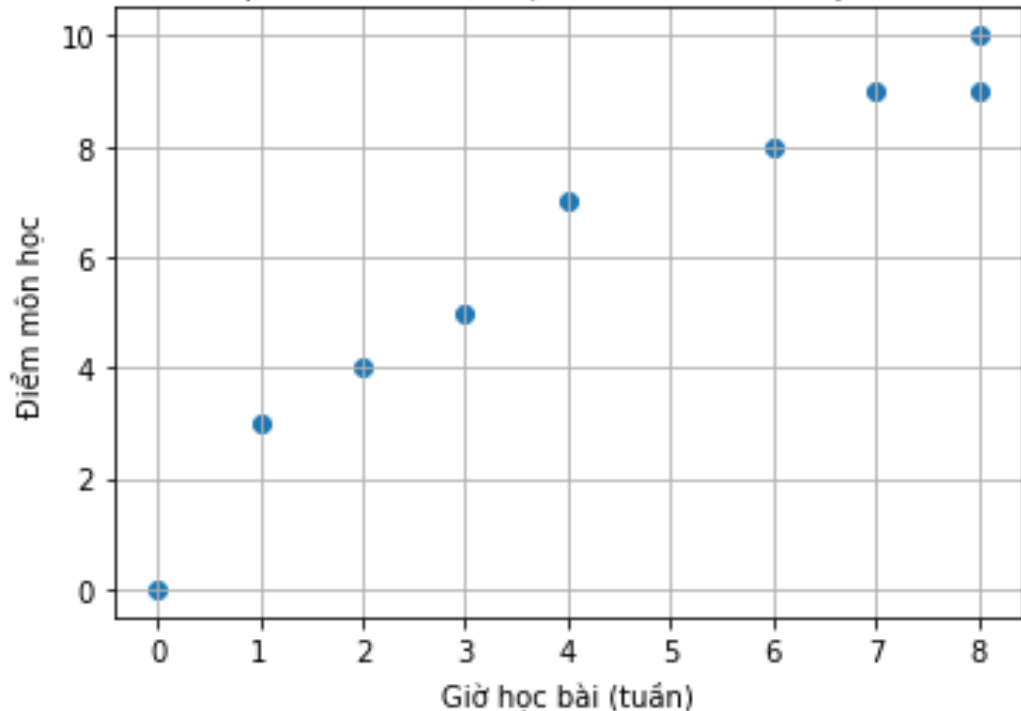
2.5 Các hàm thống kê: corrcoef

```
#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**!

BIỂU ĐỒ THỂ HIỆN MỐI TƯƠNG QUAN GIỮA GIỜ HỌC BÀI VÀ ĐIỂM THI



2.6 more...

Sinh viên tìm hiểu thêm các hàm xử lý mảng trong thư viện Numpy:

- Các hàm tính toán: multiply(), exp, sqrt, sin, cos, log ...
- Sắp xếp, lấy ma trận nghịch đảo: np.sort; a.T
- Thêm, xóa phần tử trong mảng: resize, append, insert, delete...
- Vstack, hstack, hsplit vsplit....
-

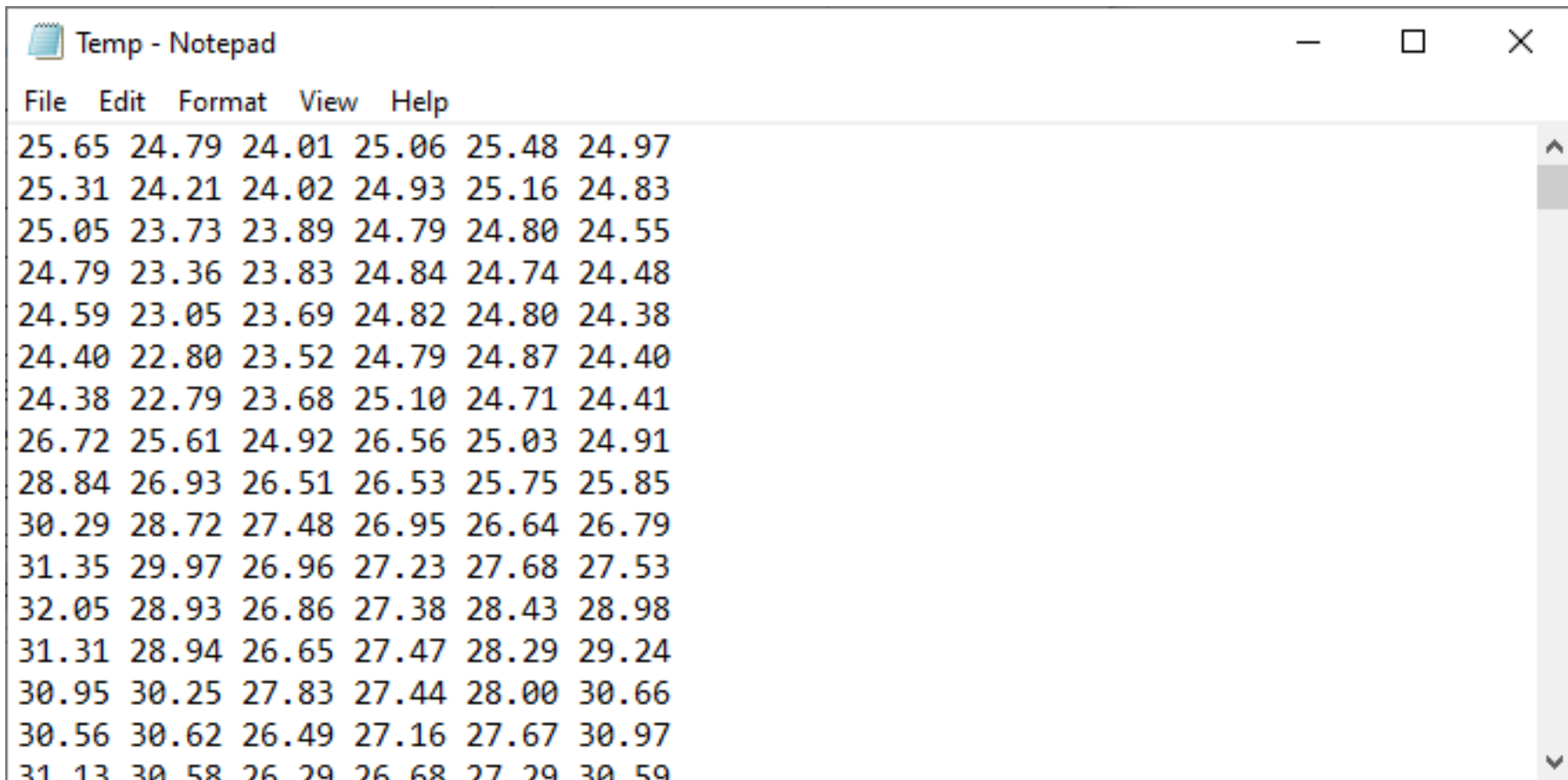
Tham khảo:

- + File: **CheatSheet-Numpy.pdf**
- + Link web: [numpy package!](#)

Thực hành

Bài 18: Làm việc với numpy

Mô tả file dữ liệu: Temp.txt



```
Temp - Notepad
File Edit Format View Help
25.65 24.79 24.01 25.06 25.48 24.97
25.31 24.21 24.02 24.93 25.16 24.83
25.05 23.73 23.89 24.79 24.80 24.55
24.79 23.36 23.83 24.84 24.74 24.48
24.59 23.05 23.69 24.82 24.80 24.38
24.40 22.80 23.52 24.79 24.87 24.40
24.38 22.79 23.68 25.10 24.71 24.41
26.72 25.61 24.92 26.56 25.03 24.91
28.84 26.93 26.51 26.53 25.75 25.85
30.29 28.72 27.48 26.95 26.64 26.79
31.35 29.97 26.96 27.23 27.68 27.53
32.05 28.93 26.86 27.38 28.43 28.98
31.31 28.94 26.65 27.47 28.29 29.24
30.95 30.25 27.83 27.44 28.00 30.66
30.56 30.62 26.49 27.16 27.67 30.97
31 13 30 58 26 29 26 68 27 29 30 59
```

Bài 18: Làm việc với numpy

Mô tả file dữ liệu: Bai18_Temp.txt

- File dữ liệu lưu trữ nhiệt độ (°C) của 6 thành phố lớn dọc theo nước Việt Nam là: Hà Nội, Vinh, Đà Nẵng, Nha trang, Hồ Chính Minh và Cà Mau
- Thời gian từ 0h ngày 15/09/2019 tới 23h ngày 22/09/2019



Bài 18: Làm việc với numpy

Mô tả file dữ liệu: Bai18_Temp.txt

Hà Nội	Vinh	Đà Nẵng	Nha Trang	HCM	Cà Mau	
25.65	24.79	24.01	25.06	25.48	24.97	Time: 0h 15/09
25.31	24.21	24.02	24.93	25.16	24.83	1h 15/09
25.05	23.73	23.89	24.79	24.80	24.55	
24.79	23.36	23.83	24.84	24.74	24.48	
24.59	23.05	23.69	24.82	24.80	24.38	
24.40	22.80	23.52	24.79	24.87	24.40	
24.38	22.79	23.68	25.10	24.71	24.41	
26.72	25.61	24.92	26.56	25.03	24.91	
28.84	26.93	26.51	26.53	25.75	25.85	
30.29	28.72	27.48	26.95	26.64	26.79	

Bài 18: Làm việc với numpy

1) Đọc dữ liệu lưu trữ trong file Bai18_Temp.txt vào biến `data_numpy`, cho biết kích thước, số chiều, kiểu dữ liệu và số phần tử của biến `data_numpy`.

```
In [16]: 1 print(data_numpy)
          2 print('-----')
          3 print('Kích thước biến:', data_numpy.shape)
          4 print('Số chiều của biến:', data_numpy.ndim)
          5 print('Kiểu dữ liệu của các phần tử:', data_numpy.dtype)
          6 print('Số phần tử:', data_numpy.size)
```

```
[[25.65 24.79 24.01 25.06 25.48 24.97]
 [25.31 24.21 24.02 24.93 25.16 24.83]
 [25.05 23.73 23.89 24.79 24.8 24.55]
 ...
 [24.81 24.47 23.4 25.86 25.05 25.29]
 [23.97 24.22 22.95 25.74 24.92 24.87]
 [22.84 23.99 22.59 25.5 24.77 24.57]]
```

```
-----
Kích thước biến: (192, 6)
Số chiều của biến: 2
Kiểu dữ liệu của các phần tử: float64
Số phần tử: 1152
```

Bài 18: Làm việc với numpy

2) Tìm nhiệt độ cao nhất (Max) – Thấp nhất (Min) – Nhiệt độ trung bình của cả 6 thành phố.

3) Tìm nhiệt độ cao nhất (Max) – Thấp nhất (Min) – Nhiệt độ trung bình của từng thành phố và hiển thị kết quả.

---THÔNG KÊ CHO CẢ 6 THÀNH PHỐ---

Nhiệt độ cao nhất: 33.45

Nhiệt độ thấp nhất: 20.93

Nhiệt độ trung bình: 26.502222222222222

1) Hà Nội

Nhiệt độ cao nhất: 33.45

Nhiệt độ thấp nhất: 21.68

Nhiệt độ trung bình: 27.712291666666667

2) Vinh (Nghệ An)

Nhiệt độ cao nhất: 32.57

Nhiệt độ thấp nhất: 22.6

Nhiệt độ trung bình: 26.719895833333336

3) Đà Nẵng

Nhiệt độ cao nhất: 29.88

Nhiệt độ thấp nhất: 20.93

Nhiệt độ trung bình: 25.522499999999997

4) Nha Trang

Nhiệt độ cao nhất: 28.68

Nhiệt độ thấp nhất: 24.5

Nhiệt độ trung bình: 26.166875000000005

5) TP Hồ Chí Minh

Nhiệt độ cao nhất: 31.06

Nhiệt độ thấp nhất: 23.22

Nhiệt độ trung bình: 26.159218749999997

6) Cà Mau

Nhiệt độ cao nhất: 31.37

Nhiệt độ thấp nhất: 23.99

Nhiệt độ trung bình: 26.732552083333333

Bài 18: Làm việc với numpy

4) Tạo một ma trận **data_thongke** gồm 3 hàng x 7 cột; các hàng lần lượt lưu trữ dữ liệu như sau:

- hàng 0: Nhiệt độ lớn nhất (Max)
- hàng 2: Nhiệt độ trung bình (Mean), làm tròn đến 2 số sau dấu phẩy
- hàng 2: Nhiệt độ nhỏ nhất (Min)

Các cột lần lượt theo thứ tự của 6 thành phố và cột cuối cùng là cột thống kê chung cho cả 6 thành phố. Lưu ra file **thongke.txt**

```
1 print(data_thongke)
2 print(type(data_thongke))
3 print('Kích thước:', data_thongke.shape)
```

```
[[33.45 32.57 29.88 28.68 31.06 31.37 33.45]
 [27.71 26.72 25.52 26.17 26.16 26.73 26.5 ]
 [21.68 22.6  20.93 24.5  23.22 23.99 20.93]]
```

```
<class 'numpy.ndarray'>
```

```
Kích thước: (3, 7)
```

