

<div> <div>A</div> <div>D</div> </div> <div> <div>A</div> <div>I</div> </div>	PHÒNG THÍ NGHIỆM THỰC HÀNH ỨNG DỤNG KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO Applied Data Science and Artificial Intelligence Lab
	 <div> LẬP TRÌNH PYTHON CHO KHOA HỌC DỮ LIỆU PYTHON FOR DATA SCIENCE </div> <div>Hà Nội ★ 11.2020</div> <div>Nothing is faster than Innovation together</div>

1

<div> <div>A</div> <div>D</div> </div> <div> <div>A</div> <div>I</div> </div>	Thông tin giảng viên
	<div>  <p>TS. TẠ QUANG CHIẾU</p> </div> <div> <p>Vị trí & Đào tạo</p> <ul style="list-style-type: none"> - Trưởng bộ môn Hệ thống thông tin và Tri thức, Khoa Công nghệ Thông tin - PhD & Post-doc in Polytech Tours, France <p>Lĩnh vực giảng dạy, nghiên cứu</p> <ul style="list-style-type: none"> - Big data, Data science, machine learning, AI - Các phương pháp cho bài toán lập kế hoạch <p>Contact:</p> <p>(E): quangchieu.ta@gmail.com</p> <p>(M): 0913 522 275</p> </div>
Dr. Tạ Quang Chiếu – (E): quangchieu.ta@gmail.com – (M): 0913 522 275	

2

<div style="display: flex; justify-content: space-between; padding: 2px;"> A D </div> <div style="display: flex; justify-content: space-between; padding: 2px;"> A I </div>	<h2 style="margin: 0;">Objectives</h2>
<ul style="list-style-type: none"> • Hiểu được tầm quan trọng của Khoa học dữ liệu • Vận dụng được các bước trong quy trình thực hiện một dự án về khoa học dữ liệu. • Kiến thức, kỹ năng cần thiết để trở thành một nhà khoa học dữ liệu • Áp dụng được Python và các thư viện phổ biến trong giải quyết một số bài toán cơ bản của Khoa học dữ liệu. 	
<div style="display: flex; justify-content: space-between;"> Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275 3/63 </div>	

3

<div style="display: flex; justify-content: space-between; padding: 2px;"> A D </div> <div style="display: flex; justify-content: space-between; padding: 2px;"> A I </div>	<h2 style="margin: 0;">References</h2>
<ul style="list-style-type: none"> [1]. Introducing Data Science [2]. Python Data Science Handbook [3]. Data Science from Scratch [4]. Python crash course [5]. Bài giảng Khoa học dữ liệu 	
<div style="display: flex; justify-content: space-between;"> Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275 4/63 </div>	

4

A A	D I	PHẦN 2: NHẬP MÔN KHOA HỌC DỮ LIỆU	
		FUNDAMENTALS OF DATA SCIENCE	
CONTENT			
1		CHAPTER 1: INTRODUCTION	
2		CHAPTER 2: PYTHON LIBRARIES FOR DATA SCIENCE	
Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275			5/63

5

A A	D I	Content of chapter 2	
2.1 Python Library for Data Science			
2.2 Numpy library			
2.3 Pandas library			
2.4 Matplotlib library			
2.5 Scikit-learn library			
Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275			6/63

6

A	D	
A	I	
<h1>NumPy Library - phần 02</h1>		
Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275		7/63

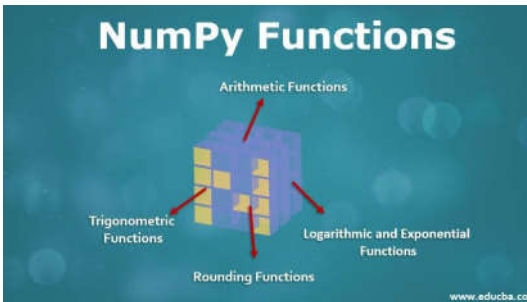
7

A	D	<h2>Nội dung phần 02</h2>
A	I	
<ul style="list-style-type: none">6. Tính toán trên mảng7. Sắp xếp các phần tử trong mảng (sort)8. Tìm kiếm các phần tử trong mảng (where)9. Ma trận vuông10. Phép toán trên 2 ma trận11. Rank(A), A^T12. Bài tập		
Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275		8/63

8

A D
A I

6. Tính toán với NumPy



NumPy Functions

Arithmetic Functions

Trigonometric Functions

Rounding Functions

Logarithmic and Exponential Functions

www.aducha.com

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275
9/63

9

A D
A I

6.1 Arithmetic operators (1)

The following table lists the arithmetic operators implemented in NumPy:

Operator	Equivalent ufunc	Description
+	<code>np.add</code>	Addition (e.g., $1 + 1 = 2$)
-	<code>np.subtract</code>	Subtraction (e.g., $3 - 2 = 1$)
-	<code>np.negative</code>	Unary negation (e.g., -2)
*	<code>np.multiply</code>	Multiplication (e.g., $2 * 3 = 6$)
/	<code>np.divide</code>	Division (e.g., $3 / 2 = 1.5$)
//	<code>np.floor_divide</code>	Floor division (e.g., $3 // 2 = 1$)
**	<code>np.power</code>	Exponentiation (e.g., $2 ** 3 = 8$)
%	<code>np.mod</code>	Modulus/remainder (e.g., $9 \% 4 = 1$)

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275
10/63

10

A

D

A

I

6.1 Arithmetic operators (2)

```

1 import numpy as np
2 x = np.arange(8)
3 print("x =", x)
4 print('-----')
5 #Các phép toán đã giới thiệu trong buổi 01
6 print("x + 5 =", x + 5)
7 print("x - 5 =", x - 5)
8 print("-x =", -x)
9 print("x * 2 =", x * 2)
10 print("x / 2 =", x / 2)
11 print("x // 2 =", x // 2)
12 print("x % 2 =", x % 2)
13 print("x ^ 3 =", x**3)

```

x = [0 1 2 3 4 5 6 7]

x + 5 = [5 6 7 8 9 10 11 12]

x - 5 = [-5 -4 -3 -2 -1 0 1 2]

-x = [0 -1 -2 -3 -4 -5 -6 -7]

x * 2 = [0 2 4 6 8 10 12 14]

x / 2 = [0. 0.5 1. 1.5 2. 2.5 3. 3.5]

x // 2 = [0 0 1 1 2 2 3 3]

x % 2 = [0 1 0 1 0 1 0 1]

x ^ 3 = [0 1 8 27 64 125 216 343]

```

1 print("x =", x)
2 print('-----')
3 #Sử dụng các phương thức của NumPy
4 print("x + 5 =", np.add(x,5))
5 print("x - 5 =", np.subtract(x, 5))
6 print("-x =", np.negative(x))
7 print("x * 2 =", np.multiply(x,2))
8 print("x / 2 =", np.divide(x,2))
9 print("x // 2 =", np.floor_divide(x, 2))
10 print("x % 2 =", np.mod(x,2))
11 print("x ^ 3 =", np.power(x,3))

```

x = [0 1 2 3 4 5 6 7]

x + 5 = [5 6 7 8 9 10 11 12]

x - 5 = [-5 -4 -3 -2 -1 0 1 2]

-x = [0 -1 -2 -3 -4 -5 -6 -7]

x * 2 = [0 2 4 6 8 10 12 14]

x / 2 = [0. 0.5 1. 1.5 2. 2.5 3. 3.5]

x // 2 = [0 0 1 1 2 2 3 3]

x % 2 = [0 1 0 1 0 1 0 1]

x ^ 3 = [0 1 8 27 64 125 216 343]

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

11/63

11

A

D

A

I

6.2 Abs | Trigonometric functions

- **np.abs() | np.absolute():**
để tính giá trị tuyệt đối của các phần tử.

```

1 x = np.array([-2, -1, 0, 1, 2])
2 print(x)
3 print('-----')
4 print(np.abs(x))
5 print(np.absolute(x))
6

```

[-2 -1 0 1 2]

[2 1 0 1 2]

[2 1 0 1 2]

Trigonometric Functions

```

1 theta = np.linspace(0, np.pi, 3)
2 print("theta =", theta)
3 print('-----')
4 print("sin(theta) =", np.sin(theta))
5 print("cos(theta) =", np.cos(theta))
6 print("tan(theta) =", np.tan(theta))
7

```

theta = [0. 1.57079633 3.14159265]

sin(theta) = [0.00000000e+00 1.00000000e+00 1.2246468e-16]

cos(theta) = [1.0000000e+00 6.123234e-17 -1.0000000e+00]

tan(theta) = [0.00000000e+00 1.63312394e+16 -1.22464680e-16]

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

12/63

12

6.3 Exponents and logarithms (1)	
Function	Description
exp(arr)	Returns exponential of an input array element wise
expm1(arr)	Returns exponential $\exp(x)-1$ of an input array element wise
exp2(arr)	Returns exponential $2^{**}x$ of all elements in an array
log(arr)	Returns natural log of an input array element wise
log10(arr)	Returns log base 10 of an input array element wise
log2(arr)	Returns log base 2 of an input array element wise
logaddexp(arr)	Returns logarithm of the sum of exponentiations of all inputs
logaddexp2(arr)	Returns logarithm of the sum of exponentiations of the inputs in base 2

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

13/63

13

6.3 Exponents and logarithms (2)	
<pre> 1 x = np.array([1, 2, 3]) 2 3 print("x =", x) 4 print('-----') 5 print("e^x =", np.exp(x)) 6 print("2^x =", np.exp2(x)) 7 print("3^x =", np.power(3, x)) </pre> <p> x = [1 2 3] ----- e^x = [2.71828183 7.3890561 20.08553692] 2^x = [2. 4. 8.] 3^x = [3 9 27] </p>	<pre> 1 x = np.array([1, 2, 4, 100]) 2 3 print("x =", x) 4 print('-----') 5 print("ln(x) =", np.log(x)) 6 print("log2(x) =", np.log2(x)) 7 print("log10(x) =", np.log10(x)) </pre> <p> x = [1 2 4 100] ----- ln(x) = [0. 0.69314718 1.38629436 4.60517019] log2(x) = [0. 1. 2. 6.64385619] log10(x) = [0. 0.30103 0.60205999 2.] </p>

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

14/63

14

np.around()


01

np.floor()

02

np.ceil()

03



6.4 Rounding Functions

Rounding Functions in NumPy

```

1 import numpy as np
2 arr = np.array([20.8999, 67.89899, 54.43409])
3
4 print(arr)
5 print('-----')
6 #1) Làm tròn tới 1 số sau dấu ,
7 print(np.around(arr,1))
8
9 #2) Làm tròn tới 2 số sau dấu ,
10 print(np.around(arr,2))
11
12 #3) Làm tròn xuống số nguyên gần nhất
13 print(np.floor(arr))
14
15 #4) Làm tròn lên số nguyên gần nhất
16 print(np.ceil(arr))

```

```

[20.8999  67.89899  54.43409]
-----
[20.9  67.9  54.4]
[20.9  67.9  54.43]
[20.  67.  54.]
[21.  68.  55.]

```

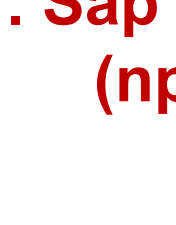
Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

15/63

15

np.sort()

01



7. Sắp xếp mảng (np.sort)

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

16/63

16

A

D

7. Sắp xếp các phần tử trong mảng (1)

np.sort()

The function name

`np.sort(`

The array you want to operate on

`a=`

The sorting algorithm you want to use to sort the data

`axis=`

The axis along which you will sort the array

`kind=`

* kind='quicksort'- Default, 'mergesort', 'heapsort', 'stable'

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

17/63

17

A

D

7. Sắp xếp các phần tử trong mảng (2)

Sắp xếp các phần tử trong một Vector

np.sort() SORTS THE VALUES OF A NUMPY ARRAY

```

1 #Sắp xếp các phần tử trong một vector
2 a = np.random.randint(1,33,15)
3
4 print('Vector ban đầu:\n', a)
5 print('-----')
6 #Sắp xếp vector a tăng dần
7 a_sort = np.sort(a)
8
9 #Sắp xếp vector a giảm dần:
10 #1) Lật vector a_sort để sắp xếp giảm dần
11 b_sort = np.flip(a_sort)
12 #2) sử dụng -np.sort(-x)
13 b_sort = -np.sort(-a)
14 print('Vector sắp xếp tăng dần: \n',a_sort)
15
16 print('Vector sắp xếp giảm dần: \n',b_sort)

```

Vector ban đầu:
[1 17 13 15 9 9 23 30 32 10 30 16 4 16 24]

Vector sắp xếp tăng dần:
[1 4 9 9 10 13 15 16 16 17 23 24 30 30 32]

Vector sắp xếp giảm dần:
[32 30 30 24 23 17 16 16 15 13 10 9 9 4 1]

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

18/63

18

7. Sắp xếp các phần tử trong mảng (3)

Sắp xếp các phần tử trong một Ma trận:

Ma trận A:

```
[[ 6 22 17 21 21 28]
 [17 22 4 19 14 9]
 [24 7 15 7 18 22]
 [ 6 9 2 8 23 3]
 [14 11 18 9 17 9]]
```

axis=1 (Sắp xếp theo hàng)

```
1 #Sắp xếp các phần tử của ma trận A
2 #2) Sắp xếp theo hàng axis=1 | Default
3 a_sort2 = np.sort(A,axis=1)
4 print('Ma trận 2:\n', a_sort2)
```

Ma trận 2:

```
[[ 6 17 21 21 22 28]
 [ 4 9 14 17 19 22]
 [ 7 7 15 18 22 24]
 [ 2 3 6 8 9 23]
 [ 9 9 11 14 17 18]]
```

axis=0 (Sắp xếp theo cột)

```
1 #1) Sắp xếp theo cột axis=0
2 a_sort1 = np.sort(A,axis=0)
3 print('Ma trận 1:\n', a_sort1)
```

Ma trận 1:

```
[[ 6 7 2 7 14 3]
 [ 6 9 4 8 17 9]
 [14 11 15 9 18 9]
 [17 22 17 19 21 22]
 [24 22 18 21 23 28]]
```

axis=None (Chuyển thành vector và sắp xếp các phần tử tăng dần theo hàng)

```
1 #3) Chuyển thành vector và sắp xếp các phần tử tăng dần theo hàng
2 v_sort = np.sort(A,axis=None)
3 print('Vector: \n', v_sort)
```

Vector:

```
[ 2  3  4  6  6  7  7  8  9  9  9  9 11 14 14 15 17 17 18 18 19 21 21
 22 22 22 23 24 28]
```

Ma trận 3:

```
[[ 2  3  4  6  6  7]
 [ 7  8  9  9  9  9]
 [11 14 14 15 17 17]
 [17 18 18 19 21 21]
 [22 22 22 23 24 28]]
```

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

19/63

19

8. Tìm kiếm (np.where)

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

20/63

20

A

D

A

I

8. Tìm kiếm trong mảng: np.where (2)

Vd1: Tìm kiếm trên ma trận:

```

1 import numpy as np
2 #Tìm kiếm trên ma trận
3 arr = np.array([(1, 2, 3, 4, 5, 4, 4),
4               (7, 3, 4, 8, 9, 6, 7)])
5 #Tìm kiếm phần tử > 4
6 x = np.where(arr > 4)
7
8 print('Ma trận A: \n',arr)
9 print('-----')
10 print(x)
11 print('Số phần tử thỏa mãn điều kiện > 4:',x[0].size)
```

Ma trận A:

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

(array([0, 1, 1, 1, 1, 1, 1], dtype=int64), array([4, 0, 3, 4, 5, 6], dtype=int64))

Số phần tử thỏa mãn điều kiện > 4: 6

Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

23/63

23

A

D

A

I

Thực hành 1

Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

24/63

24

12

A D
A I
Thực hành 1

File dữ liệu Data_BMI.txt lưu trữ thông tin chiều cao, cân nặng của 100 người. [\(Data Exercise\)](#)

Know Your BMI?

$BMI = \frac{\text{Weight in Kg}}{(\text{height in m})^2}$

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275
25/63

25

A D
A I
Thực hành 1

Yêu cầu 1.1: Đọc dữ liệu từ file Data_BMI.txt vào 2 vector tương ứng.

- v_height:** chứa dữ liệu chiều cao
- v_Weight:** Chứa dữ liệu cân nặng

Vector chiều cao:

```
[174 189 185 195 149 189 147 154 174 169 195 159 192 155 191 153 157 140
144 172 157 153 169 185 172 151 190 187 163 179 153 178 195 160 157 189
197 144 171 185 175 149 157 161 182 185 188 181 161 140 168 176 163 172
196 187 172 178 164 143 191 141 193 190 175 179 172 168 164 194 153 178
141 180 185 197 165 168 176 181 164 166 190 186 168 198 175 145 159 185
178 183 194 177 197 170 142 160 195 190]
```

Vector cân nặng:

```
[ 96 87 110 104 61 104 92 111 90 103 81 80 101 51 79 107 110 129
145 139 110 149 97 139 67 64 95 62 159 152 121 52 65 131 153 132
114 80 152 81 120 108 56 118 126 76 122 111 72 152 135 54 110 105
116 89 92 127 70 88 54 143 54 83 135 158 96 59 82 136 51 117
80 75 100 154 104 90 122 51 75 140 105 118 123 50 141 117 104 140
154 96 111 61 119 156 69 139 69 50]
```

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275
26/63

26

A D I I


Thực hành 1

?

- Yêu cầu 1.2: Tạo vector `v_height_m2` từ vector `v_height`
- theo yêu cầu sau:
 - Chuyển đổi đơn vị của các phần tử từ cm sang m
 - Tính bình phương giá trị các phần tử

Vector `v_height_m2`:

```
[3.0276 3.5721 3.4225 3.8025 2.2201 3.5721 2.1609 2.3716 3.0276 2.8561
3.8025 2.5281 3.6864 2.4025 3.6481 2.3409 2.4649 1.96 2.0736 2.9584
2.4649 2.3409 2.8561 3.4225 2.9584 2.2801 3.61 3.4969 2.6569 3.2041
2.3409 3.1684 3.8025 2.56 2.4649 3.5721 3.8809 2.0736 2.9241 3.4225
3.0625 2.2201 2.4649 2.5921 3.3124 3.4225 3.5344 3.2761 2.5921 1.96
2.8224 3.0976 2.6569 2.9584 3.8416 3.4969 2.9584 3.1684 2.6896 2.0449
3.6481 1.9881 3.7249 3.61 3.0625 3.2041 2.9584 2.8224 2.6896 3.7636
2.3409 3.1684 1.9881 3.24 3.4225 3.8809 2.7225 2.8224 3.0976 3.2761
2.6896 2.7556 3.61 3.4596 2.8224 3.9204 3.0625 2.1025 2.5281 3.4225
3.1684 3.3489 3.7636 3.1329 3.8809 2.89 2.0164 2.56 3.8025 3.61 ]
```



Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

27/63

27

A D I I

Thực hành 1


?

Yêu cầu 1.3: Tính chỉ số BMI của 100 người ngày theo công thức bên dưới, chỉ số BMI được làm tròn tới 1 số sau dấu “,”. Lưu kết quả vào vector `v_bmi`.

$$\text{BMI} = \frac{\text{Cân nặng (kg)}}{\text{Chiều cao x chiều cao (m)}}$$

Chỉ số BMI:

```
[31.7 24.4 32.1 27.4 27.5 29.1 42.6 46.8 29.7 36.1 21.3 31.6 27.4 21.2
21.7 45.7 44.6 65.8 69.9 47. 44.6 63.7 34. 40.6 22.6 28.1 26.3 17.7
59.8 47.4 51.7 16.4 17.1 51.2 62.1 37. 29.4 38.6 52. 23.7 39.2 48.6
22.7 45.5 38. 22.2 34.5 33.9 27.8 77.6 47.8 17.4 41.4 35.5 30.2 25.5
31.1 40.1 26. 43. 14.8 71.9 14.5 23. 44.1 49.3 32.4 20.9 30.5 36.1
21.8 36.9 40.2 23.1 29.2 39.7 38.2 31.9 39.4 15.6 27.9 50.8 29.1 34.1
43.6 12.8 46. 55.6 41.1 40.9 48.6 28.7 29.5 19.5 30.7 54. 34.2 54.3
18.1 13.9]
```



Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

28/63

28

A D

A I

Thực hành 1

?

Yêu cầu 1.4: Sắp xếp thứ tự các phần tử trong vector `v_bmi` theo chiều tăng dần, giảm dần.

1. Sắp xếp theo thứ tự tăng dần:

[12.8 13.9 14.5 14.8 15.6 16.4 17.1 17.4 17.7 18.1 19.5 20.9 21.2 21.3 21.7 21.8 22.2 22.6 22.7 23. 23.1 23.7 24.4 25.5 26. 26.3 27.4 27.4 27.5 27.8 27.9 28.1 28.7 29.1 29.1 29.2 29.4 29.5 29.7 30.2 30.5 30.7 31.1 31.6 31.7 31.9 32.1 32.4 33.9 34. 34.1 34.2 34.5 35.5 36.1 36.1 36.9 37. 38. 38.2 38.6 39.2 39.4 39.7 40.1 40.2 40.6 40.9 41.1 41.4 42.6 43. 43.6 44.1 44.6 44.6 45.5 45.7 46. 46.8 47. 47.4 47.8 48.6 48.6 49.3 50.8 51.2 51.7 52. 54. 54.3 55.6 59.8 62.1 63.7 65.8 69.9 71.9 77.6]

2. Sắp xếp theo thứ tự giảm dần:

[77.6 71.9 69.9 65.8 63.7 62.1 59.8 55.6 54.3 54. 52. 51.7 51.2 50.8 49.3 48.6 48.6 47.8 47.4 47. 46.8 46. 45.7 45.5 44.6 44.6 44.1 43.6 43. 42.6 41.4 41.1 40.9 40.6 40.2 40.1 39.7 39.4 39.2 38.6 38.2 38. 37. 36.9 36.1 36.1 35.5 34.5 34.2 34.1 34. 33.9 32.4 32.1 31.9 31.7 31.6 31.1 30.7 30.5 30.2 29.7 29.5 29.4 29.2 29.1 29.1 28.7 28.1 27.9 27.8 27.5 27.4 27.4 26.3 26. 25.5 24.4 23.7 23.1 23. 22.7 22.6 22.2 21.8 21.7 21.3 21.2 20.9 19.5 18.1 17.7 17.4 17.1 16.4 15.6 14.8 14.5 13.9 12.8]

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

29/63

29

A D

A I

Thực hành 1

?

Yêu cầu 1.5: Thống kê số lượng người theo từng mức dựa vào phân loại theo hình.

Tổng số: 100

1. Underweight : 10
 2. Normal : 13
 3. Overweight : 16
 4. Obese : 14
 5. Extremely Obese: 47

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

30/63

30

A

D

A

I

9. Ma trận vuông

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

31/63

31

A

D

A

I

9.1 Định thức det (A)

Định thức của ma trận vuông cấp n là tổng đại số của n! (n giai thừa) số hạng, mỗi số hạng là tích của n phần tử lấy trên các hàng và các cột khác nhau của ma trận A, mỗi tích được nhân với phần tử dấu là +1 hoặc -1 theo phép thế tạo bởi các chỉ số hàng và chỉ số cột của các phần tử trong tích

$$\det(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n a_{i, \sigma(i)}$$

Định thức của một ma trận vuông còn được viết như sau

$$\det A = \begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n} \end{vmatrix}$$

Áp dụng với các ma trận vuông cấp 1,2,3 ta có:

$$\det [a] = a$$

$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}$$

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

32/63

32

A

D

I

9.1 Định thức det (A)

`np.linalg.det(a)`: tính định thức của ma trận vuông a

```

1 import numpy as np
2 a = np.array([( 1, 3, 1, 4),
3               ( 3, 9, 5, 15),
4               ( 0, 2, 1, 1),
5               ( 0, 4, 2, 3)])
6 print('Ma trận a:\n',a)
7 det_a = np.linalg.det(a)
8 print('det(a) = ', det_a)

```

Ma trận a:

```

[[ 1  3  1  4]
 [ 3  9  5 15]
 [ 0  2  1  1]
 [ 0  4  2  3]]
det(a) = -3.999999999999999

```

```

1 import numpy as np
2 b = np.array([( 1, 2, 3, 4),
3               (-2,-1, 4, 1),
4               ( 3,-4,-5, 6),
5               ( 1, 2, 3, 4)])
6 print('Ma trận b:\n',b)
7 det_b = np.linalg.det(b)
8 print('det(b) = ', det_b)

```

Ma trận b:

```

[[ 1  2  3  4]
 [-2 -1  4  1]
 [ 3 -4 -5  6]
 [ 1  2  3  4]]
det(b) = 0.0

```

Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

33/63

33

A

D

I

9.2 Ma trận nghịch đảo

Ma trận nghịch đảo của ma trận vuông M ký hiệu M^{-1}

$M * M^{-1} = I$ (ma trận đơn vị)

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{bmatrix}$$

$$M^{-1} = \begin{bmatrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{bmatrix}$$

`np.linalg.inv(m)`: Tìm ma trận nghịch đảo của ma trận m

`det(m) = 0`: Không tồn tại ma trận nghịch đảo

Dr. Tà Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

34/63

34

A

D

9.3 Đường chéo (1)

A

I

a) Lấy phần tử trên đường chéo

a.diagonal(): trả về vector chứa các phần tử nằm trên đường chéo chính của ma trận a.

a.diagonal(k): trả về vector chứa các phần tử nằm cách đường chéo chính của ma trận a, k phần tử (k>0 trên đường chéo chính, k<0 dưới đường chéo chính)

Main diagonal

1st diagonal below main diagonal

2nd diagonal below main diagonal

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

35/63

35

A

D

9.3 Đường chéo (2)

A

I

```

1 #Lấy các phần tử nằm trên đường chéo chính
2 #của ma trận a 1 phần tử
3 d_A1 = A.diagonal(1)
4 print(d_A1)

```

```

1 #Lấy các phần tử trên đường chéo chính của
2 #ma trận vuông A
3 d_A = A.diagonal()
4 print(d_A)

```

```

1 #Lấy các phần tử nằm dưới đường chéo chính
2 #của ma trận a 4 phần tử
3 d_A1 = A.diagonal(-4)
4 print(d_A1)

```

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

36/63

36

A

D

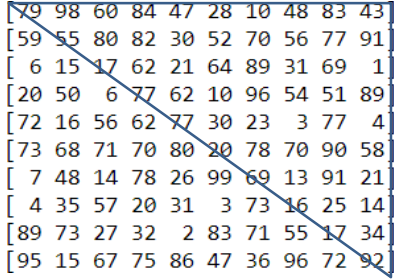
I

9.3 Đường chéo (3)

b) Ma trận tam giác (triu | tril)

np.triu(m) | np.tril(m): trả về ma trận tam giác trên | dưới của ma trận m.

np.triu(m, k): trả về ma trận trên của ma trận m cách đường chéo chính k phần tử (k = 0 (default), k<0 bên dưới đường chéo chính, k>0 bên trên đường chéo chính)



```

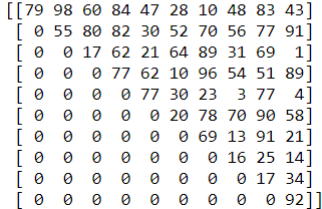
[[79 98 60 84 47 28 10 48 83 43]
 [59 55 80 82 30 52 70 56 77 91]
 [ 6 15 17 62 21 64 89 31 69  1]
 [20 50  6 77 62 10 96 54 51 89]
 [72 16 56 62 77 30 23  3 77  4]
 [73 68 71 70 80 20 78 70 90 58]
 [ 7 48 14 78 26 99 69 13 91 21]
 [ 4 35 57 20 31  3 73 16 25 14]
 [89 73 27 32  2 83 71 55 17 34]
 [95 15 67 75 86 47 36 96 72 92]]

```

```

1  #Ma trận tam giác trên của
2  #ma trận A
3  d_A1 = np.triu(A)
4  print(d_A1)

```



Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

37/63

37

A

D

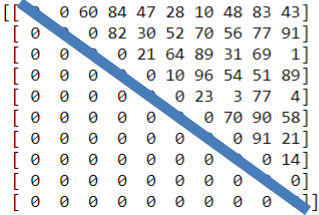
I

9.3 Đường chéo (4)

```

1  #Tạo ma trận là các phần tử trên đường chéo chính
2  #của ma trận vuông A cách đường chéo chính
3  #về phía trên 2 đường
4  d_A1 = np.triu(A,2)
5  print(d_A1)

```



```

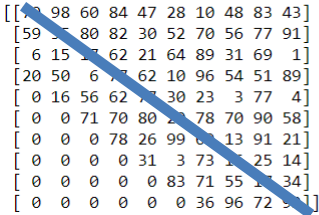
[[ 0 60 84 47 28 10 48 83 43]
 [ 0  0 82 30 52 70 56 77 91]
 [ 0  0  0 21 64 89 31 69  1]
 [ 0  0  0  0 10 96 54 51 89]
 [ 0  0  0  0  0 23  3 77  4]
 [ 0  0  0  0  0  0 70 90 58]
 [ 0  0  0  0  0  0  0 91 21]
 [ 0  0  0  0  0  0  0  0 14]
 [ 0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0]]

```

```

1  #Tạo ma trận là các phần tử trên đường chéo chính
2  #của ma trận vuông A cách đường chéo chính
3  #về phía dưới 3 đường
4  d_A1 = np.triu(A,-3)
5  print(d_A1)

```



Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

38/63

38

A D I

9.3 Đường chéo (5)

c) Vết của ma trận

Trace of a Matrix

Suppose $T \in \mathcal{L}(V)$, $F = \mathbb{C}$, and we choose a basis of V corresponding to the Decomposition Theorem. Then $\text{trace } T$ equals the sum of the diagonal entries of that matrix.

Definition: trace of a matrix

The *trace* of a square matrix A , denoted $\text{trace } A$, is defined to be the sum of the diagonal entries of A .

Example: Suppose

$$A = \begin{pmatrix} 3 & -1 & -2 \\ 3 & 2 & -3 \\ 1 & 2 & 0 \end{pmatrix}.$$

Then

$$\text{trace } A = 3 + 2 + 0 = 5.$$

Trace of AB equals trace of BA

If A and B are square matrices of the same size, then

$$\text{trace}(AB) = \text{trace}(BA).$$

```
1 #Tính trace của ma trận vuông A
2 trace_A = A.trace()
3 print('Trace of Matrix A: ', trace_A)
```

```
1 #Cách 2: Tính trace của ma trận vuông A
2 trace_A = A.diagonal().sum()
3 print('Trace of Matrix A: ', trace_A)
```

Trace of Matrix A: 519
Trace of Matrix A: 519

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275
39/63

39

A D I

9.4 Vector riêng, giá trị riêng

d) Vector riêng(Eigenvector), giá trị riêng (Eigenvalue)

Một số λ và một vector khác 0 x thỏa mãn

Eigenvector of Matrix A

$$Ax = \lambda x$$

Eigenvalue of Matrix A

được gọi lần lượt là *giá trị riêng* và *vector riêng* của A

```
1 #Tìm giá trị riêng, vector riêng của vector A
2 A = np.array([(3, 4, -2),
3               (1, 4, -1),
4               (2, 6, -1)])
5
6 eigenvalues, eigenvectors = np.linalg.eig(A)
7 print('Eigenvalues: ', eigenvalues)
8 print('Eigenvectors: \n', eigenvectors)
```

Eigenvalues: [3. 2. 1.]

Eigenvectors:

```
[[-4.08248290e-01  2.67561446e-15  7.07106781e-01]
 [-4.08248290e-01  4.47213595e-01 -3.33066907e-16]
 [-8.16496581e-01  8.94427191e-01  7.07106781e-01]]
```

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275
40/63

40

A

D

A

I

Thực hành 2

Dr. Tà Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

41/63

41

A

D

A

I

Thực hành 2

?

Yêu cầu 2.1: Sử dụng vector_weight trong bài thực hành 1 chuyển về ma trận kích thước 10 x 10.

Vector Weight:

```
[ 96 87 110 104 61 104 92 111 90 103 81 80 101 51 79 107 110 129
145 139 110 149 97 139 67 64 95 62 159 152 121 52 65 131 153 132
114 80 152 81 120 108 56 118 126 76 122 111 72 152 135 54 110 105
116 89 92 127 70 88 54 143 54 83 135 158 96 59 82 136 51 117
80 75 100 154 104 90 122 51 75 140 105 118 123 50 141 117 104 140
154 96 111 61 119 156 69 139 69 50]
```

Ma trận Weight:

```
[[ 96 87 110 104 61 104 92 111 90 103]
[ 81 80 101 51 79 107 110 129 145 139]
[110 149 97 139 67 64 95 62 159 152]
[121 52 65 131 153 132 114 80 152 81]
[120 108 56 118 126 76 122 111 72 152]
[135 54 110 105 116 89 92 127 70 88]
[ 54 143 54 83 135 158 96 59 82 136]
[ 51 117 80 75 100 154 104 90 122 51]
[ 75 140 105 118 123 50 141 117 104 140]
[154 96 111 61 119 156 69 139 69 50]]
```

Dr. Tà Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

42/63

42

A D

A I

Thực hành 2

?

Yêu cầu 2.2: Cho biết ma trận weight có tồn tại ma trận nghịch đảo không? Nếu có hãy xác định ma trận weight⁻¹ ?

Yêu cầu 2.3: Tạo vector_diagonal chứa các phần tử trên đường chéo chính của ma trận weight, tính trace của ma trận weight.

a) Đường chéo chính của ma trận weight:
[96 80 97 131 126 89 96 90 104 50]

b) Trace của ma trận weight: 959

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

43/63

43

A D

A I

Thực hành 2

?

Yêu cầu 2.4: Tìm giá trị lớn nhất của ma trận đường chéo trên và ma trận đường chéo dưới không bao gồm các phần tử nằm trên đường chéo chính của ma trận weight?

```
[[ 0 87 110 104 61 104 92 111 90 103]
 [ 0 0 101 51 79 107 110 129 145 139]
 [ 0 0 0 139 67 64 95 62 159 152]
 [ 0 0 0 0 153 132 114 80 152 81]
 [ 0 0 0 0 0 76 122 111 72 152]
 [ 0 0 0 0 0 0 92 127 70 88]
 [ 0 0 0 0 0 0 0 59 82 136]
 [ 0 0 0 0 0 0 0 0 122 51]
 [ 0 0 0 0 0 0 0 0 0 140]
 [ 0 0 0 0 0 0 0 0 0 0]]
```

Phần tử max: 159

```
[[ 0 0 0 0 0 0 0 0 0 0]
 [ 81 0 0 0 0 0 0 0 0 0]
 [110 149 0 0 0 0 0 0 0 0]
 [121 52 65 0 0 0 0 0 0 0]
 [120 108 56 118 0 0 0 0 0 0]
 [135 54 110 105 116 0 0 0 0 0]
 [ 54 143 54 83 135 158 0 0 0 0]
 [ 51 117 80 75 100 154 104 0 0 0]
 [ 75 140 105 118 123 50 141 117 0 0]
 [154 96 111 61 119 156 69 139 69 0]]
```

Phần tử max: 158

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

44/63

44

A
D

A
I

10. Phép toán với 2 ma trận

Dr. Tà Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

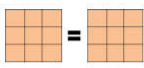
45/63

45

A
D


A
I

10. Phép toán trên 2 ma trận




Equality of Matrices

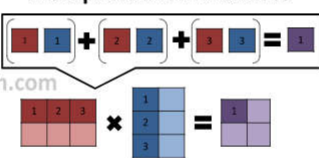
Addition of Matrices

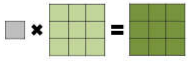


Subtraction of Matrices



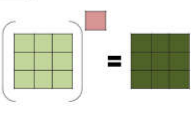
Multiplication of Matrices





Matrix Multiplied by a Scalar

Power of a Matrix



Dr. Tà Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

46/63

46

A D
A I

10. Phép toán trên 2 ma trận

a) So sánh 2 ma trận

np.equal(a,b) | ==: trả về ma trận (T|F) so sánh từng phần tử của ma trận a và ma trận b theo vị trí.

Lưu ý: ma trận a, b có cùng kích thước (m,n)

Matrix a:

```
[[ 9  4 19  1 18]
 [15 11  1  9 14]
 [17  8  4 10 13]]
```

Matrix b:

```
[[ 6  4  9 12  4]
 [ 3  6 11 14 10]
 [ 1  6  5 12  2]]
```

```
1 #1) So sánh 2 ma trận
2 equal_ab = np.equal(a,b)
3 #hoặc equal_ab = a==b
4
5 print(equal_ab)
```

```
[[False  True False False False]
 [False False False False False]
 [False False False False False]]
```

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

47/63

47

A D
A I

10. Phép toán trên 2 ma trận

b) Cộng, trừ 2 ma trận

np.add(a,b) | +: trả về ma trận có các phần tử là tổng của phần tử của ma trận a và ma trận b.

np.subtract(a,b) | -: trả về ma trận có các phần tử là hiệu của phần tử ma trận a và ma trận b

Lưu ý: ma trận a, b có cùng kích thước (m,n)

```
1 #Phép cộng 2 ma trận
2 sum_ab = np.add(a,b)
3 #hoặc sum_ab = a + b
4 print (sum_ab)
```

```
[[15  8 28 13 22]
 [18 17 12 23 24]
 [18 14  9 22 15]]
```

```
1 #Phép trừ 2 ma trận
2 sub_ab = np.subtract(a,b)
3 #hoặc sub_ab = a - b
4 print (sub_ab)
```

```
[[ 3  0 10 -11 14]
 [12  5 -10  -5  4]
 [16  2  -1  -2 11]]
```

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

48/63

48

A
D

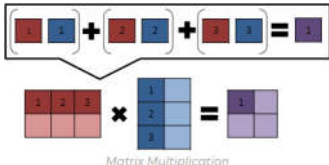
A
I

10. Phép toán trên 2 ma trận

c) Nhân 2 ma trận

np.dot(a,b) | @: trả về ma trận kết quả là tích của 2 ma trận a,b

Lưu ý: ma trận a có kích thước (m,n)
ma trận c có kích thước (n,k)
ma trận ac có kích thước (m,k)



Matrix a:

```
[[ 9  4 19  1 18]
 [15 11  1  9 14]
 [17  8  4 10 13]]
```

Matrix c:

```
[[13  8  9 13]
 [17 11  4  3]
 [13  7  2 18]
 [12  1  7  2]
 [ 1 13  6  4]]
```

```
1 #3) Tích của 2 ma trận:
2 multi_ac = np.dot(a,c)
3 #hoặc multi_ac1 = a@c
4 print(multi_ac)
```

```
[[462 484 250 545]
 [517 439 328 320]
 [542 431 341 389]]
```

```
1 #Tích của 2 vector:
2 vector_a = np.random.randint(1,20,10)
3 vector_b = np.random.randint(1,20,10)
4 #Thực hiện tính tích của 2 vector
5 #Kết quả trả về một số
6 vector_ab = vector_a @ vector_b
7
8 print('Vector a:\n',vector_a)
9 print('Vector b:\n',vector_b)
10 print('Tích của hai vector:\n',vector_ab)
```

Vector a:
[7 15 18 6 3 13 8 11 2 4]

Vector b:
[4 1 15 12 13 16 17 6 9 14]

Tích của hai vector:
908

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

49/63

49

A
D

A
I

Thực hành 3

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

50/63

50

A

D

A

I

Thực hành 3

Yêu cầu 3.1: Chuyển vector_height, vector_weight trong bài thực hành 1 về ma trận height, weight kích thước 10 x 10.

Thực hiện các phép toán: So sánh; Cộng; Trừ; Nhân hai ma trận height và weight

Ma trận height:

```
[[174 189 185 195 149 189 147 154 174 169]
[195 159 192 155 191 153 157 140 144 172]
[157 153 169 185 172 151 190 187 163 179]
[153 178 195 160 157 189 197 144 171 185]
[175 149 157 161 182 185 188 181 161 140]
[168 176 163 172 196 187 172 178 164 143]
[191 141 193 190 175 179 172 168 164 194]
[153 178 141 180 185 197 165 168 176 181]
[164 166 190 186 168 198 175 145 159 185]
[178 183 194 177 197 170 142 160 195 190]]
```

Ma trận weight:

```
[[ 96 87 110 104 61 104 92 111 90 103]
[ 81 80 101 51 79 107 110 129 145 139]
[110 149 97 139 67 64 95 62 159 152]
[121 52 65 131 153 132 114 80 152 81]
[120 108 56 118 126 76 122 111 72 152]
[135 54 110 105 116 89 92 127 70 88]
[ 54 143 54 83 135 158 96 59 82 136]
[ 51 117 80 75 100 154 104 90 122 51]
[ 75 140 105 118 123 50 141 117 104 140]
[154 96 111 61 119 156 69 139 69 50]]
```

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

51/63

51

A

D

A

I

11. Hạng của ma trận, ma trận chuyển vị

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

52/63

52

A

D

A

I

11.1 Hạng của ma trận A

Hạng của ma trận là cấp cao nhất của định thức con khác 0 của ma trận đó.

Hạng của ma trận A kí hiệu $\text{rank}(A)$ hoặc $r(A)$

- + Ma trận 0 có hạng bằng 0
- + Ma trận A cấp $m \times n$ thì $0 \leq r(A) \leq \min(m, n)$
- + Ma trận A vuông cấp n :
 - Nếu $\det(A) \neq 0$ thì $r(A) = n$
 - Nếu $\det(A) = 0$ thì $r(A) < n$

`np.linalg.matrix_rank(A)`: Tính hạng của ma trận A

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

53/63

53

A

D

A

I

11.1 Hạng của ma trận A

Find the rank and nullity of the matrix

$$A = \begin{bmatrix} -1 & 2 & 0 & 4 & 5 & -3 \\ 3 & -7 & 2 & 0 & 1 & 4 \\ 2 & -5 & 2 & 4 & 6 & 1 \\ 4 & -9 & 2 & -4 & -4 & 7 \end{bmatrix}$$

Solution.

The reduced row-echelon form of A is

$$\begin{bmatrix} 1 & 0 & -4 & -28 & -37 & 13 \\ 0 & 1 & -2 & -12 & -16 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Since there are two nonzero rows, the row space and column space are both two-dimensional, so $\text{rank}(A)=2$.

```

1 A=np.array([(-1, 2, 0, 4, 5,-3),
2             ( 3,-7, 2, 0, 1, 4),
3             ( 2,-5, 2, 4, 6, 1),
4             ( 4,-9, 2,-4,-4, 7)])
5 #Tìm hạng của ma trận A
6 rank_a = np.linalg.matrix_rank(A)
7 print(A)
8 print('Rank(A) = ', rank_a)

```

```

[[-1  2  0  4  5 -3]
 [ 3 -7  2  0  1  4]
 [ 2 -5  2  4  6  1]
 [ 4 -9  2 -4 -4  7]]
Rank(A) = 2

```

```

1 #Hạng của ma trận 0
2 A_0 = np.zeros((4,5))
3 print(A_0)
4 rank = np.linalg.matrix_rank(A_0)
5 print('Rank(A_0) = ', rank)

```

```

[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
Rank(A_0) = 0

```

Ma trận B1:

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

$\det(B1) = 0.0$
 $\text{Rank}(B1) = 3$

Ma trận B2:

$$\begin{bmatrix} 1 & 3 & 1 & 4 \\ 3 & 9 & 5 & 15 \\ 0 & 2 & 1 & 1 \\ 0 & 4 & 2 & 3 \end{bmatrix}$$

$\det(B2) = -4.0$
 $\text{Rank}(B2) = 4$

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

54/63

54

A

D

11.2 Ma trận chuyển vị

Chuyển vị của ma trận $m \times n$ **A** là ma trận $n \times m$ **A^T** tạo ra bằng cách chuyển hàng thành cột và cột thành hàng:

A.T: Tìm ma trận chuyển vị của ma trận A

```

1 A=np.array([(-1, 2, 0, 4, 5,-3),
2             ( 3,-7, 2, 0, 1, 4),
3             ( 2,-5, 2, 4, 6, 1),
4             ( 4,-9, 2,-4,-4, 7)])
5 #Tìm ma trận chuyển vị của A
6 A_T = A.T
7 print('Ma trận A:\n',A)
8 print('Ma trận chuyển vị của A:\n',A_T)

```

```

1 B =np.array([( 1, 3, 1, 4),
2             ( 3, 9, 5,15),
3             ( 0, 2, 1, 1),
4             ( 0, 4, 2, 3)])
5 #Tìm ma trận chuyển vị của B
6 B_T = B.T
7 print('Ma trận B:\n',B)
8 print('Ma trận chuyển vị của B:\n',B_T)

```

Ma trận A:

```

[[-1  2  0  4  5 -3]
 [ 3 -7  2  0  1  4]
 [ 2 -5  2  4  6  1]
 [ 4 -9  2 -4 -4  7]]

```

Ma trận chuyển vị của A:

```

[[-1  3  2  4]
 [ 2 -7 -5 -9]
 [ 0  2  2  2]
 [ 4  0  4 -4]
 [ 5  1  6 -4]
 [-3  4  1  7]]

```

Ma trận B:

```

[[ 1  3  1  4]
 [ 3  9  5 15]
 [ 0  2  1  1]
 [ 0  4  2  3]]

```

Ma trận chuyển vị của B:

```

[[ 1  3  0  0]
 [ 3  9  2  4]
 [ 1  5  1  2]
 [ 4 15  1  3]]

```

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

55/63

55

A

D

Thực hành 4

Thực hành 4

Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

56/63

56

A D I I

Thực hành 4

Yêu cầu 4.1: Học viên tìm ma trận nghịch đảo và hạng của ma trận Height, Weight trong bài thực hành số 3:

Ma trận nghịch đảo height.T:

```
[[174 195 157 153 175 168 191 153 164 178]
[189 159 153 178 149 176 141 178 166 183]
[185 192 169 195 157 163 193 141 190 194]
[195 155 185 160 161 172 190 180 186 177]
[149 191 172 157 182 196 175 185 168 197]
[189 153 151 189 185 187 179 197 198 170]
[147 157 190 197 188 172 172 165 175 142]
[154 140 187 144 181 178 168 168 145 160]
[174 144 163 171 161 164 164 176 159 195]
[169 172 179 185 140 143 194 181 185 190]]
```

Hạng của ma trận height: 10

Ma trận nghịch đảo weight.T:

```
[[ 96  81 110 121 120 135  54  51  75 154]
[ 87  80 149  52 108  54 143 117 140  96]
[110 101  97  65  56 110  54  80 105 111]
[104  51 139 131 118 105  83  75 118  61]
[ 61  79  67 153 126 116 135 100 123 119]
[104 107  64 132  76  89 158 154  50 156]
[ 92 110  95 114 122  92  96 104 141  69]
[111 129  62  80 111 127  59  90 117 139]
[ 90 145 159 152  72  70  82 122 104  69]
[103 139 152  81 152  88 136  51 140  50]]
```

Hạng của ma trận weight: 10

Dr. Tà Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

57/63

57

A D I I

Thực hành 4

Yêu cầu 4.2: Học viên thực hành và trả lời các câu hỏi sau:

- 1

The rank of matrix $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \\ 3 & 3 & 3 \end{bmatrix}$ is equal to

A) 4 B) 3 C) 2 D) 1
- 2

If $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$ and $\det(A)=0$ then rank of a matrix A is

A) Greater than or equal to 3 B) Strictly less than 3
C) Less than or equal to 3 D) Strictly greater than 3 .
- 3

The rank of matrix $\begin{bmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix}$ is

A) 10 B) 5 C) 2 D) 1.

Dr. Tà Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275

58/63

58

A
D
A
I
Thực hành 4

4 For matrix A of order $m \times n$, the rank r of matrix A is
 A) $r \geq \min\{m, n\}$ B) $r \geq \max\{m, n\}$
 C) $r \leq \min\{m, n\}$ D) $r \leq \max\{m, n\}$

5 A 5×7 matrix has all its entries equal to -1, then rank of matrix is
 A) 7 B) 5 C) 1 D) zero

6 The rank of the following matrix by determinant method $\begin{bmatrix} 2 & 3 & 4 \\ 4 & 3 & 1 \\ 1 & 2 & 4 \end{bmatrix}$ is
 A) 2 B) 3 C) 1 D) 0

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275
59/63

59

A
D
A
I

BÀI TẬP

Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275
60/63

60

A D		Bài tập	
A	I		
<p>Bài tập 2.3. Tạo một ma trận 4x4 toàn các giá trị False</p> <p>Bài tập 2.4. Cho một dãy số nguyên 100 phần tử, hãy tách lấy tất cả những phần tử lẻ cho vào một mảng</p> <p>Bài tập 2.5. Cho một dãy số tự nhiên 20 phần tử, hãy thay thế tất cả những phần tử lẻ bằng số -1</p> <p>Bài tập 2.6. Hai mảng a và b có cùng số dòng, hãy ghép chúng theo các dòng thành mảng c, các cột của a rồi đến các cột của b</p> <p>Bài tập 2.7. Mảng a và b có cùng số cột, hãy ghép chúng theo các cột thành mảng c, các dòng của a rồi đến của b</p>			
Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275			61/63

61

A D		Bài tập	
A	I		
<p>Bài tập 2.8. Cho một mảng a, hãy in ra tất cả những phần tử trong khoảng từ 5 đến 10</p> <p>Bài tập 2.9. Sinh ra một mảng số thực có 1000 phần tử, các phần tử nằm trong khoảng từ -0.5 đến < 0.5</p> <p>Bài tập 2.10. Sinh một ma trận 3x5 gồm các số ngẫu nhiên từ 0 đến nhỏ hơn 10, tính và in ra số lớn nhất trên mỗi dòng của ma trận</p> <p>Bài tập 2.11. Nhập mảng a và b có 10 phần tử, tính khoảng cách euclid giữa a và b</p>			
Dr. Tạ Quang Chiếu – [E]: quangchieu.ta@gmail.com – [M]: 0913 522 275			62/63

62



Dr. Tạ Quang Chiếu - [E]: quangchieu.ta@gmail.com - [M]: 0913 522 275

63/63