



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Лабораторна робота №1
з дисципліни
Аналіз даних з використанням мови Python

Виконав:

студент групи ІІІ-13:
Ал Хадам М.Р.

Перевірила:

ст. викладач
Тимофєєва Ю.С.

Київ 2023

Мета роботи: Ознайомитись з основними можливостями роботи з масивами бібліотеки NumPy, визначити статистичні характеристики.

Завдання до лабораторної роботи

Створити програму, яка:

1. Генерує випадкові і невипадкові масиви різними способами, зазначеними в теоретичних відомостях.

```
In 1 1 import numpy as np
      2
      3
      4 arr = np.arange(15, 50, 3)
      5 arr
      Executed at 2023.09.19 13:11:22 in 164ms
```

Out 1 ▾

÷	0 ÷
0	15
1	18
2	21
3	24
4	27
5	30
6	33
7	36
8	39
9	42

2. Масив з десяти цілочислених одиниць

```
In 2 1 arr = np.ones(10, dtype=int)
      2 arr
      Executed at 2023.09.19 13:11:22 in 11ms
```

Out 2 ▾

÷	0 ÷
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1

3. Масив двовимірний, кожний рядок якого містить п'ять нулів, тип даних з плаваючою точкою

```
In 3 1 arr = np.zeros ((2, 5), dtype=float)
      2 arr
```

Executed at 2023.09.19 13:11:22 in 91ms

Out 3 ▾

	0	1	2	3	4
0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0

4. Масив з семи значень, розміщених рівномірно між 50 та 100.

```
In 4 1 arr = np.linspace(50, 100, 7)
      2 arr
```

Executed at 2023.09.19 13:11:22 in 80ms

Out 4 ▾

	0
0	50.000000
1	58.333333
2	66.666667
3	75.000000
4	83.333333
5	91.666667
6	100.000000

5. Масив 5x6 з рівномірно розподілених випадкових чисел від 0 до 1.

```
In 5 1 arr = np.random.random((5, 6))
      2 arr
```

Executed at 2023.09.19 13:11:22 in 63ms

Out 5 ▾

	0	1	2	3	4	5
0	0.870711	0.002020	0.856161	0.444974	0.091009	0.226558
1	0.740492	0.842197	0.117680	0.660174	0.498881	0.684209
2	0.939846	0.564562	0.520356	0.843581	0.131636	0.336928
3	0.195545	0.350178	0.629386	0.092705	0.860696	0.490252
4	0.501985	0.541262	0.732830	0.776906	0.761385	0.428163

6. Масив 5x5 випадкових цілих чисел від 1 до 10.

```
In 6 1 arr = np.random.randint(1, 10, (5, 5))
      2 arr
```

Executed at 2023.09.19 13:11:22 in 52ms

Out 6 ▾

	0	1	2	3	4
0	2	3	8	8	4
1	3	6	4	2	5
2	9	6	6	5	6
3	3	1	5	2	4
4	8	4	8	2	1

7. Порожній масив з 7 елементів. Насправді, його елементами буде випадковий вміст комірок пам'яті

```
In 7 1 arr = np.empty(7)
      2 arr
```

Executed at 2023.09.19 13:11:22 in 32ms

Out 7 7 rows x 1 columns np.ndarray

	0
0	50.000000
1	58.333333
2	66.666667
3	75.000000
4	83.333333
5	91.666667
6	100.000000

2. Демонструє звернення до елементів масиву за допомогою індексів, в тому числі від'ємних; виділення підмасивів як одновимірних, так і багатовимірних масивів.

Завдання 2

```
In 8 1 arr = np.array(
      2     [[1, 2, 3],
      3       [4, 5, 6],
      4       [7, 8, 9]]
      5 )
      6 arr.shape
```

Executed at 2023.09.19 13:11:22 in 21ms

Out 8 (3, 3)

Демонструє звернення до елементів масиву за допомогою індексів, в тому числі від'ємних;

```
In 9 1 print(arr[1])
      2 print(arr[-1])
      3 print(arr[0, 0])
```

Executed at 2023.09.19 13:11:22 in 18ms

4 5 6
7 8 9
1

Демонструє виділення підмасивів як одновимірних, так і багатовимірних масивів.

```
In 10 1 print(arr[1:])
      2 print(arr[1:, :2])
```

Executed at 2023.09.19 13:11:22 in 133ms

4 5 6
7 8 9
4 5
7 8

3. Демонструє основні арифметичні операції над масивами, а також роботу методів `reduce`, `accumulate`, `outer`.

Завдання №3

```
In 11 1 a = np.ones(4)
      2 a
```

Executed at 2023.09.19 13:11:22 in 118ms

Out 11 ▾

	<	<	4 rows ▾	>	>	4 rows × 1 columns	np.ndarray ↗
÷			0 ÷				
0			-6.0				
1			-6.0				
2			-6.0				
3			-6.0				

Демонструє основні арифметичні операції над масивами

```
In 12 1 a += 3
      2 print(a)
      3 a -= 10
      4 print(a)
      5 a *= 2
      6 print(a)
      7 a /= 2
      8 print(a)
      9 a = a ** 2
     10 print(a)
     11 a %= 19
     12 print(a)
     13 a = np.negative(a)
     14 print(a)
```

Executed at 2023.09.19 13:11:22 in 150ms

▾

```
[4.  4.  4.  4.]
[-6. -6. -6. -6.]
[-12. -12. -12. -12.]
[-6. -6. -6. -6.]
[36. 36. 36. 36.]
[17. 17. 17. 17.]
[-17. -17. -17. -17.]
```

А також роботу методів reduce, accumulate, outer.

```
In 13 1 a = np.arange(0, 6)
      2 print(a)
      3 np.add.reduce(a)
      Executed at 2023.09.19 13:11:22 in 134ms
```

[0 1 2 3 4 5]

Out 13 15

```
In 14 1 np.add.accumulate(a)
      Executed at 2023.09.19 13:11:22 in 118ms
```

Out 14 ▾ |< < 6 rows ▾ > >| 6 rows × 1 columns np.ndarray ↗

÷	0 ÷
0	0
1	1
2	3
3	6
4	10
5	15

```
In 15 1 a = np.arange(1, 10)
      2 print(a)
      3 np.multiply.outer(a, a)
      Executed at 2023.09.19 13:11:22 in 104ms
```

[1 2 3 4 5 6 7 8 9]

Out 15 ▾ |< < 9 rows ▾ > >| 9 rows × 9 columns np.ndarray ↗

÷	0 ÷	1 ÷	2 ÷	3 ÷	4 ÷	5 ÷	6 ÷	7 ÷	8 ÷
0	1	2	3	4	5	6	7	8	9
1	2	4	6	8	10	12	14	16	18
2	3	6	9	12	15	18	21	24	27
3	4	8	12	16	20	24	28	32	36
4	5	10	15	20	25	30	35	40	45
5	6	12	18	24	30	36	42	48	54
6	7	14	21	28	35	42	49	56	63
7	8	16	24	32	40	48	56	64	72
8	9	18	27	36	45	54	63	72	81

4. Вираховує статистичні характеристики, а саме, мінімальне і максимальне значення, вибіркове середнє, дисперсію, середньоквадратичне відхилення, медіану та 25 та 75 персентилі, величини ширина пелюстки (petal_width) з набору даних щодо квіток ірису (iris.csv).

Завдання №4

Виразовує статистичні характеристики, а саме, мінімальне і максимальне значення, вибіркові середнє, дисперсію, середньоквадратичне відхилення, медіану та 25 та 75 перцентилі, величини ширина пелюстки (petal_width) з набору даних щодо квіток ірису (iris.csv).

```
In 16 1 import pandas as pd
2
3
4 iris = pd.read_csv('iris.csv')
5 petal = np.array(iris['petal_width'])
6 iris
```

Executed at 2023.09.19 13:11:23 in 400ms

Out 16 ▾

150 rows × 5 columns `pd.DataFrame`

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

```
In 17 1 iris.describe()
Executed at 2023.09.19 13:11:23 in 42ms
```

Out 17 ▾

8 rows × 4 columns `pd.DataFrame`

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
In 18 1 print('Min value of petal_width:', np.min(petal))
2 print('Max value of petal_width:', np.max(petal))
3
4 print('\nSum of petal_width values:', np.sum(petal))
5 print('Mean value of petal_width:', np.mean(petal))
6
7 print('\nStandard deviation of petal_width:', np.std(petal))
8 print('Variance of petal_width:', np.var(petal))
9
10 print('\nThe 25th percentile of petal_width:', np.percentile(petal, 25))
11 print('Median value of petal_width:', np.median(petal))
12 print('The 75th percentile of petal_width:', np.percentile(petal, 75))
```

Executed at 2023.09.19 13:11:23 in 25ms

Min value of petal_length: 0.1
Max value of petal_length: 2.5

Sum of petal_length values: 179.8
Mean value of petal_length: 1.1986666666666668

Standard deviation of petal_length: 0.7606126185881716
Variance of petal_length: 0.5785315555555555

The 25th percentile of petal_length: 0.3
Median value of petal_length: 1.3
The 75th percentile of petal_length: 1.8