



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

Лабораторна робота №1
з дисципліни Аналіз даних з використанням мови Python
Базове знайомство з бібліотекою NumPy

Виконав:
студент групи ІА-11:
Никифоров М. С.

Перевірила:
ст. викладач
Тимофєєва Ю.С.

Київ 2023

Завдання 1

Згенерувати випадкові і невипадкові масиви різними способами, зазначеними в теоретичних відомостях:

- `np.array(data)`
- `np.arange(1,16,2)`
- `np.ones/zeros/empty`
- `np.linspace(0, 1, 5)`
- `np.random.random((4, 3))`
- `np.random.randint(0, 100, (4, 4))`

Всі масиви мають наступні атрибути: `ndim`(розмірність), `shape`(розмір кожного виміру), `size`(загальний розмір масиву), `dtype`(тип елементів), `nbytes`(розмір масиву в байтах), `itemsize` (розмір елементу в байтах).

```
import numpy as np
print("Simple example:\n", np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]))
print("Arange example:", np.arange(0, 23, 2))
print("Ones/Zeros/Empty example:\n", np.ones((2, 4, 5), dtype="int"))
print("Linspace example:", np.linspace(2, 7, 6, dtype="int"))
print("Random example:\n", np.random.random((2, 4)).round(2))
print("Random Integer example:\n", np.random.randint(10, 40, (2, 4)))
```

Simple example:

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

Arange example: [0 2 4 6 8 10 12 14 16 18 20 22]

Ones/Zeros/Empty example:

```
[[[1 1 1 1 1]
  [1 1 1 1 1]
  [1 1 1 1 1]
  [1 1 1 1 1]]]
```

```
[[1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]
 [1 1 1 1 1]]]
```

Linspace example: [2 3 4 5 6 7]

Random example:

```
[[0.73 0.06 0.72 0.48]
 [0.15 0.03 0.86 0.36]]
```

Random Integer example:

```
[[25 24 32 15]
 [33 39 20 12]]
```

Завдання 2

Продемонструвати:

- звернення до елементів масиву за допомогою індексів
- виділення одновимірних підмасивів
- виділення багатовимірних підмасивів

Підмасиви NumPy не є копіями даних масиву, а їх представленням. Тобто якщо змінити дані в підмасиві, то вони зміняться й у початковому масиві.

```
import numpy as np
arr_1 = np.arange(0, 10, 1)
arr_2 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

print("Fourth element of arr_1 ---", arr_1[4])
print("Last element of arr_1 ---", arr_1[-1])
print("Middle element of arr_2 ---", arr_2[1, 1], "\n")

print("Subarray of arr_1 ---", arr_1[0:-2])
print("Subarray of arr_2 ---\n", arr_2[0:2, 1:3])
```

```
Fourth element of arr_1 --- 4
Last element of arr_1 --- 9
Middle element of arr_2 --- 5
```

```
Subarray of arr_1 --- [0 1 2 3 4 5 6 7]
Subarray of arr_2 ---
[[2 3]
 [5 6]]
```

Завдання 3

Продемонструвати:

- основні арифметичні операції над масивами
- роботу методів reduce, accumulate, outer

Додавання (+, np.add), віднімання (-, np.subtract), множення (*, np.multiply), ділення (/, np.divide), піднесення до ступеню(**, np.power), ділення за модулем(% , np.mod), зміна знаку на протилежний (-, np.negative).

```
import numpy as np
arr_1 = np.arange(0, 10, 1, dtype="float")

arr_1+=7
print("Add 7 to array = ", arr_1)

arr_1-=7
print("Subtract 7 from array = ", arr_1)

arr_1*=2
print("Multiply array by 2 = ", arr_1)

arr_1/=2
print("Divide array by 2 = ", arr_1)

arr_1**=2
print("Power array by 2 = ", arr_1)

arr_1%=7
print("Mod array by 7 = ", arr_1)

arr_1 = -arr_1
print("Negative array = ", arr_1)

print("Sum of the getting array ---", np.add.reduce(arr_1))
print("Sum(by steps) of the getting array ---", np.add.accumulate(arr_1))
print("Sum(of all elements) of the getting array ---\n", np.add.outer(arr_1, arr_1))
```

Add 7 to array = [7. 8. 9. 10. 11. 12. 13. 14. 15. 16.]

Subtract 7 from array = [0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]

Multiply array by 2 = [0. 2. 4. 6. 8. 10. 12. 14. 16. 18.]

Divide array by 2 = [0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]

Power array by 2 = [0. 1. 4. 9. 16. 25. 36. 49. 64. 81.]

Mod array by 7 = [0. 1. 4. 2. 2. 4. 1. 0. 1. 4.]

Negative array = [-0. -1. -4. -2. -2. -4. -1. -0. -1. -4.]

Sum of the getting array --- -19.0

Sum(by steps) of the getting array --- [-0. -1. -5. -7. -9. -13. -14. -14. -15. -19.]

Sum(of all elements) of the getting array ---

```
[[ -0. -1. -4. -2. -2. -4. -1. -0. -1. -4.]
 [-1. -2. -5. -3. -3. -5. -2. -1. -2. -5.]
 [-4. -5. -8. -6. -6. -8. -5. -4. -5. -8.]
 [-2. -3. -6. -4. -4. -6. -3. -2. -3. -6.]
 [-2. -3. -6. -4. -4. -6. -3. -2. -3. -6.]
 [-4. -5. -8. -6. -6. -8. -5. -4. -5. -8.]
 [-1. -2. -5. -3. -3. -5. -2. -1. -2. -5.]
 [-0. -1. -4. -2. -2. -4. -1. -0. -1. -4.]
 [-1. -2. -5. -3. -3. -5. -2. -1. -2. -5.]
 [-4. -5. -8. -6. -6. -8. -5. -4. -5. -8.]]
```

Завдання 4

Вирахувати статистичні характеристики величини ширина пелюстки з набору даних щодо квіток ірису:

- мінімальне і максимальне значення
- вибіркове середнє
- дисперсію
- середньоквадратичне відхилення
- медіану
- 25 та 75 персентилі

Сума значень `np.sum`, мінімальне та максимальне значення `np.min`, `np.max`, середнє значення `np.mean`, середньоквадратичне відхилення `np.std`, дисперсія `np.var`, медіана `np.median`, персентилі `np.percentile`

```
import numpy as np
import pandas as pd

iris_data = pd.read_csv("Data_Iris.csv")
petal_width = np.array(iris_data['petal_width'])
print("Max value = ", petal_width.max())
print("Min value = ", petal_width.min())
print("Selective average = ", np.mean(petal_width).round(2))
print("Statistical dispersion = ", np.var(petal_width).round(2))
print("Standard deviation = ", np.std(petal_width).round(2))
print("Median value = ", np.median(petal_width).round(2))
print("25 percentile = ", np.percentile(petal_width, 25).round(2))
print("75 percentile = ", np.percentile(petal_width, 75).round(2))
```

```
Max value =  2.5
Min value =  0.1
Selective average =  1.2
Statistical dispersion =  0.58
Standard deviation =  0.76
Median value =  1.3
25 percentile =  0.3
75 percentile =  1.8
```

Висновки:

В ході виконання лабораторної роботи було виконано ряд завдань, які допомогли ознайомитися з основними функціями і можливостями бібліотеки NumPy. Основні висновки з кожного завдання наведено нижче:

1. Були створені випадкові та невипадкові масиви за допомогою NumPy, використовуючи різні методи і функції, які надає бібліотека. Це дозволило зрозуміти, як створювати і маніпулювати масивами даних.

2. Було показано, як звертатися до окремих елементів масивів за допомогою індексів, включаючи від'ємні індекси. Також було показано, як виділяти підмасиви з одновимірних і багатовимірних масивів, що є важливим для подальших операцій з даними.

3. Завдяки NumPy було продемонстровано виконання основних арифметичних операцій над масивами, такі як додавання, віднімання, множення і ділення. Також були вивчені методи `reduce`, `accumulate` і `outer` для операцій над масивами, що дозволило розширити можливості обробки даних.

4. Для набору даних із файлу `iris.csv` було виконано обчислення різних статистичних характеристик, таких як мінімальне і максимальне значення, вибіркове середнє, дисперсія, середньоквадратичне відхилення, медіана та квартилі. Це дозволило отримати інформацію про розподіл даних і визначити їх основні характеристики.