

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Ймовірнісні основи програмної інженерії»

Лабораторна робота № 2

Виконав:	Мельничук Дмитро Олегович	Перевірів:	Вечерковська А.С.
Група	ІПЗ-24(1)	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Лінійне перетворення та Графічне зображення даних

Мета – навчитись використовувати на практиці набуті знання про лінійні перетворення та графічне зображення даних.

Завдання 1

1. Знайдіть $Q(1)$, $Q(3)$ та $P(90)$.
2. Знайдіть середнє та стандартне відхилення цих оцінок.
3. Через незадоволення низькими оцінками викладач вирішив використати шкалу форми $y = ax + b$, щоб відредагувати оцінки. Він хотів, щоб середнє значення масштабних оцінок становило 95, а оцінка 100, щоб залишалася рівною 100.
4. Показати дані за допомогою діаграми "стовбур – листя".
5. Відобразити дані за допомогою коробкового графіка.

Математична модель:

Завдання 1:

Для знаходження першого квартилю необхідно знайти 25 персантиль, для 3-го квартилю – 75 персантиль, та знайти 90 персантиль за формулою:

$$n = \frac{k}{100} (N + 1)$$

Де k – номер персантилю

N – кількість елементів

n – номер елементу, який є персантилем

Таким чином ми знайдемо номер елемента який i є персантилем, далі для знаходження значення використаємо формулу:

$$P_n = x_n + x_{\text{дроб}} * (x_{n+1} - x_n)$$

Де:

P_n - значення n -ого персантилю

x_n – Значення елемента з вибірки, який стоїть за номером цілої частини n

$x_{\text{дроб}}$ – значення дробової частини n

x_{n+1} – Значення елемента з вибірки, який стоїть за номером цілої частини $n + 1$

Завдання 2:

Для знаходження середнього відхилення використаємо формулу:

$$\frac{1}{n} \sum_{i=1}^n |x_i - m(X)|$$

Де:

n – кількість елементів

x_i – поточний елемент

$m(x)$ – середнє значення з вибірки

Для знаходження стандартного відхилення використаємо формулу:

$$s_x = \sqrt{s_x^2(x)}$$

$$s_x^2 = \frac{\sum_{x \in X} f_i \cdot (x_i - \bar{x})^2}{N - 1}$$

Де:

N - кількість елементів

x_i – поточний елемент

\bar{x} – середнє значення з вибірки

S_x – стандарте відхилення

Завдання 3:

Спочатку знайдем значення a та b розв'язавши систему:

$$\begin{cases} \text{Max} = \text{Max} * a + b \\ \text{Aver} = \bar{x} * a + b \end{cases}$$

Де:

Max – Значення максимальної оцінки

Aver - значення бажаної середньої оцінки

\bar{x} – середнє значення оцінок

a і b – значення невідомих, які необхідно знайти

Тепер підставимо a і b в формулу:

$$y = ax + b$$

та знайдемо значення нових оцінок

де:

x – значення поточної оцінки

Завдання 4:

Щоб створити діаграму стовбур листя необхідно знайти максимальний та мінімальний елемент, потім перший стовбчик заповнити значеннями цих елементів починаючи з мінімального, і закінчуючи максимальним, але підставляються вони з кроком в 10 та береться тільки 1 частина числа, у другий стовчик ставиться 2-га частина числа, причому не тільки першого, а й всіх чисел з однаковою першою частиною

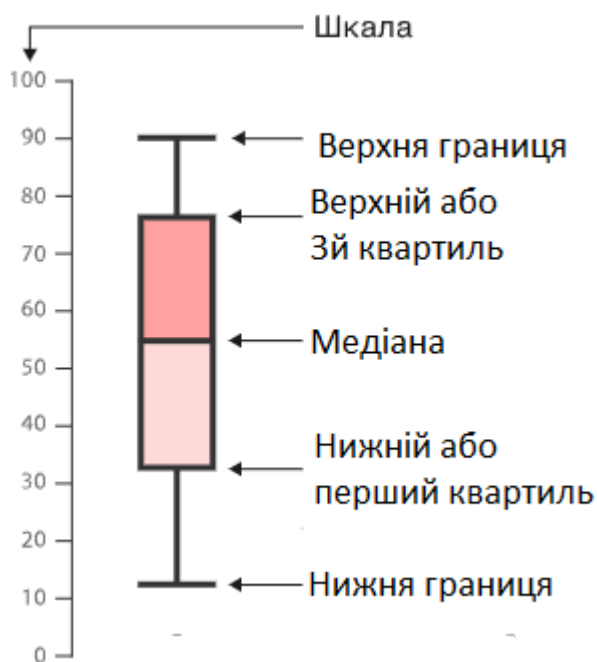
Приклад діаграми для вибірки 44 46 47 49 63 64 66 68 68 72 72 75 76 81 84 88 106 :

```
4 | 4 6 7 9
5 |
6 | 3 4 6 8 8
7 | 2 2 5 6
8 | 1 4 8
9 |
10 | 6
ключ: 6|3=63
```

Також необхідно знайти ключ. Ключем буде значення першого елемента, після якого кожному наступному елементу в 1 стовчику відповідає масив значень у другому

Завдання 5:

Для того аби побудувати коробковий графік необхідно знайти: медіану, перший та третій квартилі, а також верхню та нижню границі, які є максимальним та мінімальним значенням вибірки



Код алгоритму:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import solve
data = [0]
res = open("result.txt", "w+")

def connect_txt(data):
    nameoffile = input()
    data = np.genfromtxt(nameoffile, dtype='int')
    data = np.delete(data, 0)
    return data

def diagramlist(i):
    print("\nList diagram")
    print("-----")

    res.write("\n\nList diagram")
    res.write("\n-----")

    key = 0
    keycount = 0
    while i <= max(data):
        mas = []
        for j in range(len(data)):
            if i < data[j] < i + 10:
                if key == 0:
                    key = data[j]
                    mas.append(data[j] % 10)
            elif data[j] == i:
                if key == 0:
                    key = i
                    mas.append(0)

        if not mas and keycount < 2:
            key = 0
        else:
            keycount += 1

        print(str(i / 10) + " \t| " + str(mas))

        res.write("\n"+str(i / 10) + " \t| " + str(mas))
        i += 10
    print("Key = " + str(key))

    res.write("\nKey = " + str(key))

def pfunc(pindex):
    index = pindex * (count + 1) - 1
    Percentile = data[int(index)] + (index % int(index)) * (data[int(index) + 1] - data[int(index)])
    return Percentile

def standartdeviation():
    sum = 0
    totalSum = 0
    for i in range(len(data)):
        sum += data[i]
    midleX = sum / len(data)

    for i in range(len(data)):
        totalSum += (data[i] - midleX)**2
```

```

result = np.sqrt(totalSum/(len(data)-1))

res.write("\nStandart deviation = " + str(result))

return result

def middledeviation():
    sum = 0
    totalSum = 0
    for i in range(len(data)):
        sum += data[i]
    midleX = sum / len(data)
    for i in range(len(data)):
        totalSum += abs(data[i] - midleX)

    result = (totalSum/(len(data)))

    res.write("\nMiddle deviation = " + str(result))

    return result

def lineal():
    sum = 0
    result = []
    for i in data:
        sum += i
    midle = sum / len(data)

    a = np.array([
        [100*1, 1, ],
        [1*midle, 1, ]
    ])
    x = solve(a, np.array([100, 95]))
    print("\na = " + str(x[0]) + "\nb = " + str(x[1]))
    for i in range(len(data)):
        result.append(x[0]*data[i]+x[1])
    print("\nResult marks: " + str(result))

    res.write("\na = " + str(x[0]) + "\nb = " + str(x[1]))
    res.write("\nResult marks: " + str(result))

def boxdiagram():
    plt.boxplot(data)
    plt.grid()
    plt.show()

'''----- MAIN -----'''

data = connect_txt(data)
print("Enter data = " + str(data))
data = sorted(data)
print("Sorted data = " + str(data))
count = len(data)

res.write("\nEnter data = " + str(data))
res.write("\nSorted data = " + str(data))

Q1 = pfunc(1/4)
Q3 = pfunc(3/4)
P90 = pfunc(0.9)

```

```

print("\nFirst quartile = ", Q1)
print("\nThird quartile = ", Q3)
print("\n90's percentile = ", P90)

res.write("\nFirst quartile = " + str(Q1))
res.write("\nThird quartile = " + str(Q3))
res.write("\n90's percentile = " + str(P90))

minimal = min(data)
maximal = max(data)

print("\nStandart deviation = ", standartdeviation())
print("\nMiddle deviation = ", middledeviation())

lineal()

diagramlist(min(data))
res.close()
boxdiagram()

```

Результат на консолі та випробовування алгоритму:

Для 10 елементів:

```

Enter 10 int
Enter data = [ 40  65  62  70 100  90  66  70  95  84]
Sorted data = [40, 62, 65, 66, 70, 70, 84, 90, 95, 100]

First quartile =  64.25

Third quartile =  91.25

90's percentile =  99.5

Standart deviation =  18.103406677565783

Middle deviation =  14.440000000000001

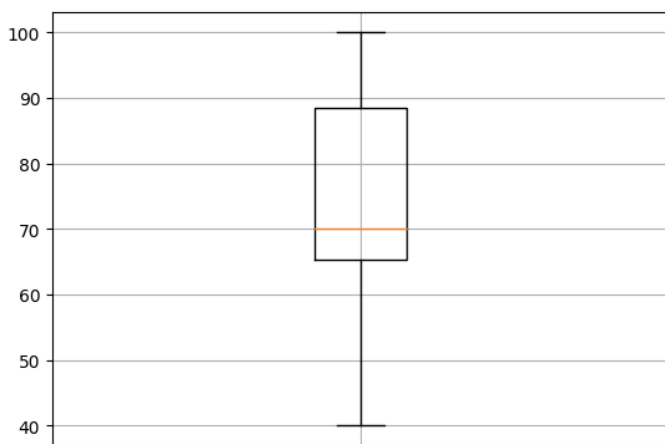
a = 0.19379844961240322
b = 80.62015503875968

Result marks: [88.3720930232558, 92.63565891472868, 93.21705426356588, 93.41085271317829, 94.18604651162791, 94.18604651162791, 96.89922480620154, 98.06201550387597, 99.03100775193798, 100.0]

List diagram
-----
4.0 | [0]
5.0 | []
6.0 | [2, 5, 6]
7.0 | [0, 0]
8.0 | [4]
9.0 | [0, 5]
10.0 | [0]
Key = 62

```

Графік:

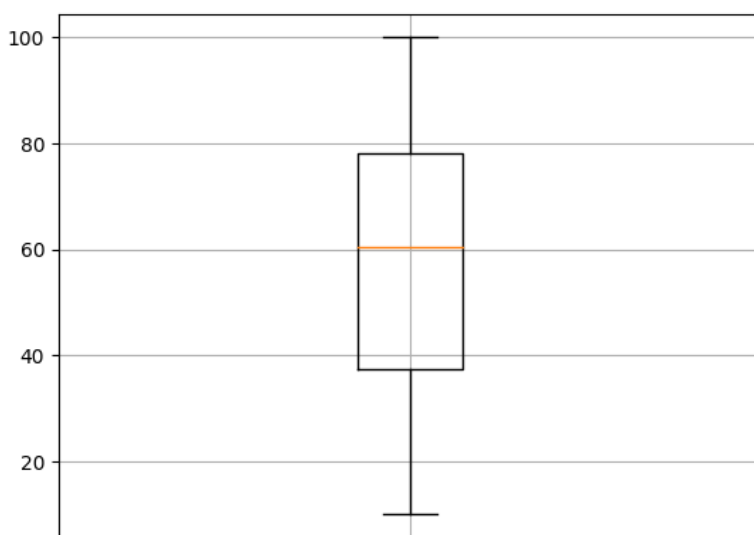


Графік я будував за допомогою бібліотеки rpyplot, і для його будування функція не запитує значень а будує тільки за допомогою вхідних даних. Але можна перевірити правильність графіку просто подивившись на всі його параметри та порівняти з даними, які виводилися до нього такі, як Q_1 , Q_3 максимальне та мінімальне значення вибірки

Для 100 елементів:

```
Enter data = [ 64  51  97  52  46  99  31  99  88  46  79  54  36  82  71  28  51  25
 42  80  65  22  81  16  68  85  12  58  77  51  56  73  76  83  44  41
 78  35  15  25  66  25  81  38  69  61  92  36  68  57  10  79  53  16
 19  19  60  91  92  67  56  50  94  77  26  47  61  68  22  22  82  55
 58  22  97  38  70  61  87  82  77  83  14  87 100  73  76  14  70  45
 49 100  63  61  35  98  78  22  35  60]
Sorted data = [10, 12, 14, 14, 15, 16, 16, 19, 19, 22, 22, 22, 22, 22, 25, 25, 25, 26, 28, 31, 35, 35, 35, 3
First quartile =  36.5
Third quartile =  78.75
90's percentile =  91.9
Standart deviation =  25.280615003387382
Middle deviation =  21.243199999999999
a = 0.1189343482397716
b = 88.10656517602284
Result marks: [89.29590865842056, 89.5337773549001, 89.77164605137965, 89.77164605137965, 89.89058039961941,
List diagram
-----
1.0      | [0, 2, 4, 4, 5, 6, 6, 9, 9]
2.0      | [2, 2, 2, 2, 2, 5, 5, 5, 6, 8]
3.0      | [1, 5, 5, 5, 6, 6, 8, 8]
4.0      | [1, 2, 4, 5, 6, 6, 7, 9]
5.0      | [0, 1, 1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 8]
6.0      | [0, 0, 1, 1, 1, 1, 3, 4, 5, 6, 7, 8, 8, 8, 9]
7.0      | [0, 0, 1, 3, 3, 6, 6, 7, 7, 7, 8, 8, 9, 9]
8.0      | [0, 1, 1, 2, 2, 2, 3, 3, 5, 7, 7, 8]
9.0      | [1, 2, 2, 4, 7, 7, 8, 9, 9]
10.0     | [0, 0]
Key = 10
```

Графік:



Також значення розрахунків записуються у текстовий файл:

```
Enter data = [10, 12, 14, 14, 15, 16, 16, 19, 19, 22, 22, 22]
Sorted data = [10, 12, 14, 14, 15, 16, 16, 19, 19, 22, 22, 22]
First quartile = 36.5
Third quartile = 78.75
90's percentile = 91.9
Standart deviation = 25.280615003387382
Middle deviation = 21.243199999999999
a = 0.1189343482397716
b = 88.10656517602284
Result marks: [89.29590865842056, 89.53377735490041, 89.77777777777777]

List diagram
-----
1.0      | [0, 2, 4, 4, 5, 6, 6, 9, 9]
2.0      | [2, 2, 2, 2, 2, 5, 5, 5, 6, 8]
3.0      | [1, 5, 5, 5, 6, 6, 8, 8]
4.0      | [1, 2, 4, 5, 6, 6, 7, 9]
5.0      | [0, 1, 1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 8]
6.0      | [0, 0, 1, 1, 1, 1, 3, 4, 5, 6, 7, 8, 8, 8, 9]
7.0      | [0, 0, 1, 3, 3, 6, 6, 7, 7, 7, 8, 8, 9, 9]
8.0      | [0, 1, 1, 2, 2, 2, 3, 3, 5, 7, 7, 8]
9.0      | [1, 2, 2, 4, 7, 7, 8, 9, 9]
10.0     | [0, 0]
Key = 10
```

Висновок:

Навчився використовувати на практиці набуті знання про лінійні перетворення, а саме знаходження кuartилів, персантилів, стандартне та середнє відхилення. Також навчився будувати коробкові графіки, а також діаграми "стовбур – листя".