

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
Лінійне перетворення та Графічне зображення даних

Виконав:	Зарожевський Євген Сергійович	Перевірила:	Марцафей А.С.
Група	ІПЗ-22	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Мета роботи:

Навчитись використовувати на практиці набуті знання про лінійні перетворення та графічне зображення даних.

Завдання:

1. Знайдіть Q_1, Q_3, 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.
6. Зробити висновок.

Побудова математичної моделі:

Для визначення кватилів та персентилів використовувалась формула:

$$P_n = \frac{1}{n} * (N + 1) \text{ th value};$$

Для визначення середнього та стандартного відхилень використовувались формули:

$$s_x = \sqrt{Var(x)}$$

$$a_x = \frac{\sum_{i=1}^n |x_i - Mean|}{n}$$

Для зміни середнього значення за лінійною трансформацією використовувалась формула:

$$y = ax + b,$$

$$\text{Де } \begin{cases} 100 = 100a + b \\ \text{New Average} = Mean + b \end{cases}$$

Для побудови коробкового графіку використовувалась бібліотека matplotlib мови програмування python.

Код алгоритму:

```
import math

import matplotlib.pyplot as plt
import numpy as np

fileName = input("Enter name of file without extension: ")

f = open(f'{fileName}.txt', 'r')

length = int(f.readline())

data = []

for i in range(length):
    data.append(int(f.readline()))

f.close()

data.sort()

f = open('output.txt', 'w')

f.write(f'Data: {data}\n')

q1IntPart = int((length + 1)/4)
q1FloatPart = ((length + 1)/4) % 1

q1 = data[q1IntPart - 1] + q1FloatPart * (data[q1IntPart] - data[q1IntPart - 1])

f.write(f'Q1 = {data[q1IntPart - 1]} + {q1FloatPart} * ({data[q1IntPart]} - {data[q1IntPart - 1]}) = {q1}\n')

q3IntPart = int(3 * (length + 1) / 4)
q3FloatPart = (3 * (length + 1)/4) % 1

q3 = data[q3IntPart - 1] + q3FloatPart * (data[q3IntPart] - data[q3IntPart - 1])

f.write(f'Q3 = {data[q3IntPart - 1]} + {q3FloatPart} * ({data[q3IntPart]} - {data[q3IntPart - 1]}) = {q3}\n')

p90IntPart = int(9 * (length + 1) / 10)
p90FloatPart = round((9 * (length + 1) / 10) % 1, 2)

p90 = data[p90IntPart - 1] + p90FloatPart * (data[p90IntPart] - data[p90IntPart - 1])

f.write(f'P90 = {data[p90IntPart - 1]} + {p90FloatPart} * ({data[p90IntPart]} - {data[p90IntPart - 1]}) = {p90}\n')

total = 0

for i in range(len(data)):
    total += data[i]

mean = total / len(data)

averageDeviation = 0
```

```

standardDeviation = 0

for i in range(len(data)):
    averageDeviation += math.fabs(data[i] - mean)
    standardDeviation += math.pow(data[i] - mean, 2)

averageDeviation /= len(data)
standardDeviation /= len(data)
standardDeviation = math.sqrt(standardDeviation)

averageDeviation = round(averageDeviation, 2)
standardDeviation = round(standardDeviation, 2)

f.write(f'Average deviation = {averageDeviation}\n')
f.write(f'Standard deviation = {standardDeviation}\n')

eq = np.array([[100, 1], [mean, 1]])
res = np.array([100, 95])
[a, b] = np.linalg.solve(eq, res)

newData = []
newTotal = 0

for i in range(len(data)):
    newData.append(round(data[i] * a + b, 2))
    newTotal += newData[i]

f.write(f'New data with average mark equals to 95:\n')
f.write(f'Formula: y = {a} * x + {b}\n')
f.write(f'{newData}\nNew Average = {round(newTotal/len(newData), 2)}\n')

stemAndLeafDictionary = {}
for i in range(len(data)):
    stringMark = str(data[i])
    if stringMark[:-1] in stemAndLeafDictionary:
        if stringMark[0] == '1' and stringMark[1] == '0' and len(stringMark) ==
3:
            key = f'{stringMark[0]}{stringMark[1]}'
            stemAndLeafDictionary[key].append(stringMark[2:])
        else:
            stemAndLeafDictionary[stringMark[0]].append(stringMark[1:])
    else:
        if stringMark[0] == '1' and stringMark[1] == '0' and len(stringMark) ==
3:
            key = f'{stringMark[0]}{stringMark[1]}'
            stemAndLeafDictionary[key] = [stringMark[2:]]
        else:
            stemAndLeafDictionary[stringMark[0]] = [stringMark[1:]]

f.write('Stem\t\t\t\t\tLeaf\n')
for key in stemAndLeafDictionary:
    f.write(f'{key}\t\t\t\t\t{stemAndLeafDictionary[key]}\n')

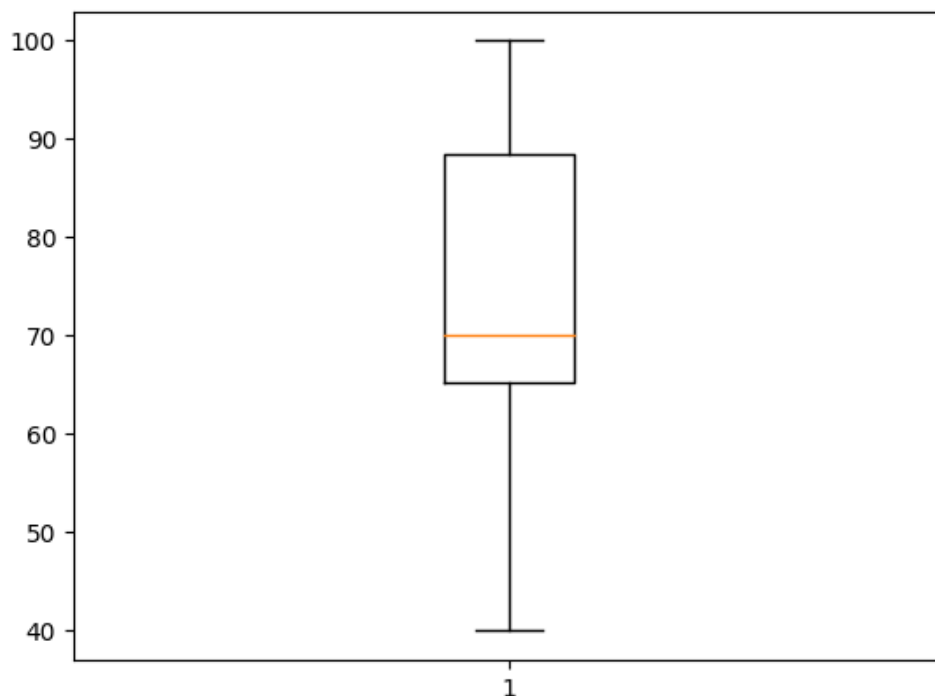
plt.boxplot(data)
plt.savefig('boxplot')
plt.show()

```

Випробування алгоритму:

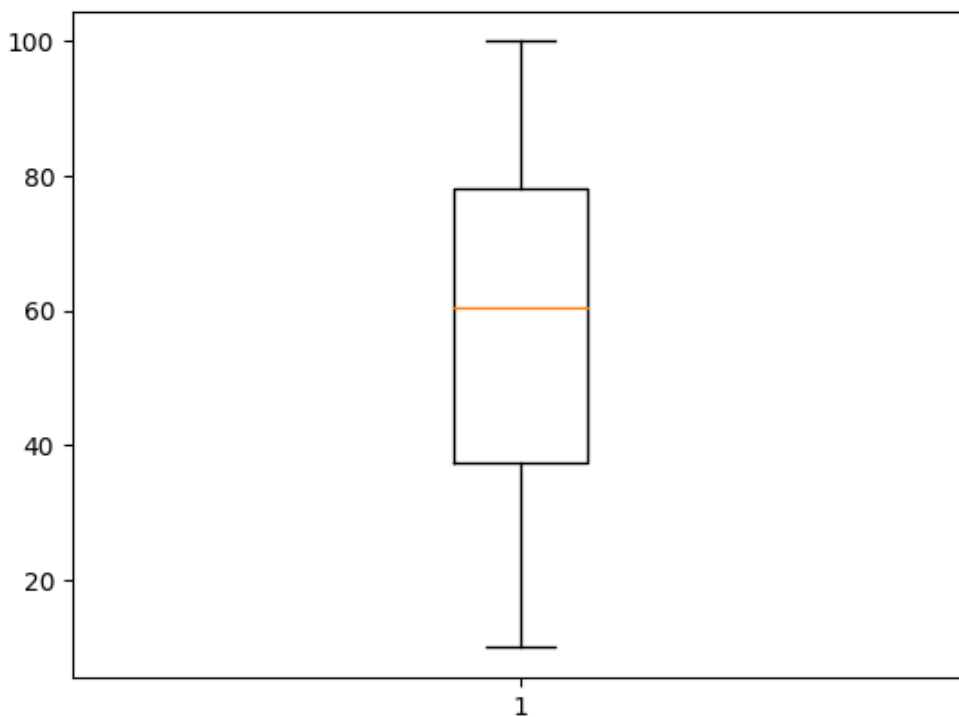
Результат роботи програми при вхідних даних розміром 10:

```
Data: [40, 62, 65, 66, 70, 70, 84, 90, 95, 100]
Q1 = 62 + 0.75 * (65 - 62) = 64.25
Q3 = 90 + 0.25 * (95 - 90) = 91.25
P90 = 95 + 0.9 * (100 - 95) = 99.5
Average deviation = 14.44
Standard deviation = 17.17
New data with average mark equals to 95:
Formula:  $y = 0.19379844961240308 * x + 80.62015503875969$ 
[88.37, 92.64, 93.22, 93.41, 94.19, 94.19, 96.9, 98.06, 99.03, 100.0]
New Average = 95.0
Stem    |   Leaf
4        ['0']
6        ['2', '5', '6']
7        ['0', '0']
8        ['4']
9        ['0', '5']
10       ['0']
```



Результат роботи програми при вхідних даних розміром 100:

```
Data: [10, 12, 14, 14, 15, 16, 16, 19, 19, 22, 22, 22, 22, 22, 25, 25, 25, 26, 28, 31, 35, 35, 35, 36, 36, 38, 38, 41, 42, 44, 45, 46, 46, 47, 49, 50, 51, 52]
Q1 = 36 + 0.25 * (38 - 36) = 36.5
Q3 = 78 + 0.75 * (79 - 78) = 78.75
P90 = 91 + 0.9 * (92 - 91) = 91.9
Average deviation = 21.24
Standard deviation = 25.15
New data with average mark equals to 95:
Formula:  $y = 0.1189343482397716 * x + 88.10656517602284$ 
[89.3, 89.53, 89.77, 89.77, 89.89, 90.01, 90.01, 90.37, 90.37, 90.72, 90.72, 90.72, 90.72, 90.72, 91.08, 91.08, 91.08, 91.2, 91.44, 91.79, 92.27, 92.27, 92.27]
New Average = 95.0
Stem | Leaf
1    ['0', '2', '4', '5', '6', '6', '9', '9']
2    ['2', '2', '2', '2', '5', '5', '5', '6', '8']
3    ['1', '5', '5', '5', '6', '6', '8', '8']
4    ['1', '2', '4', '5', '6', '6', '7', '9']
5    ['0', '1', '1', '1', '2', '3', '4', '5', '6', '6', '7', '8', '8']
6    ['0', '0', '1', '1', '1', '1', '3', '4', '5', '6', '7', '8', '8', '8', '9']
7    ['0', '0', '1', '3', '3', '6', '6', '7', '7', '7', '8', '8', '9', '9']
8    ['0', '1', '1', '2', '2', '2', '3', '3', '5', '7', '7', '8']
9    ['1', '2', '2', '4', '7', '7', '8', '9', '9']
10   ['0', '0']
```



Висновок:

Виконано завдання другої лабораторної роботи. Опановані навички використання на практиці набутих знань про лінійні перетворення та графічне зображення даних. Розроблена програма знаходить перший та третій кuartиль, 90-ий персентиль, змінює розподіл з використанням формули $y = ax + b$, відображає дані за допомогою діаграми «стовбур-листя» та коробкового графіку.