

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:	Бойко Костянтин Богданович	Перевірила:	
Група	ІПЗ-24(1)	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		

2022

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

Завдання

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 відредувати 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

1. Для розрахунку 1-ого квартилю(25 персантиль), 3-ого(75 персантилю) та 90-ого персантилю будемо використовувати формулу:

$$n = k \cdot 100 (N + 1), \text{ де}$$

k – номер персантилю

N – кількість елементів

n – номер елементу, який є персантилем

Так ми знайшли номер елемента(індекс), який є персантилем. Тепер щоб знайти його значення нам потрібно використати наступну формулу:

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

P_n - значення n -ого персантилю

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

Хдроб – значення дробової частини n

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

2. Потрібно знайти *середнє відхилення*. Для цього 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

Також щоб знайти *стандартне відхилення* використаємо:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - x_{\text{ср}})^2}{n - 1}}$$

, де

n - кількість елементів

x_i – поточний елемент

$x_{\text{ср}}$ – середнє значення з вибірки

S – стандартне відхилення

3. Для виконання завдання потрібно знайти значення a та b , розв'язавши систему рівнянь

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

, де

Max – Значення максимальної оцінки

Aver - значення бажаної середньої оцінки

\bar{x} – середнє значення оцінок

a і b – значення невідомих, які необхідно знайти

Тепер щоб змінити значення оцінок нам потрібно підставити a і b в формулу:

$$y = ax + b$$

де: x – значення поточної оцінки

y – значення оцінки в кінцевому результаті

4. Для створення діаграми *стовбур-листя* потрібно знайти максимальний і мінімальні елементи, після чого перший стовбчик заповнюємо значеннями цих елементів від мінімального до максимального. Зауважимо, що елементи підставляються з кроком в 10, береться тільки 1 частина числа у перший стовбчик та 2 частина числа у другий стовбчик, однак не тільки одного, а й усіх чисел в яких рівна перша частина.

Приклад того, як буде виглядати діаграма:

Стебла	Листки
0	5 8
1	2 3 5 7
2	0 0 0 5 8 8 9
3	0 0 1 3 3 3 6 6 7 7 7 7 7 8 8 8 8 9 9
4	1 3 5 5 5 6 7 7 8 8 8 8 9 9
5	0 0 0 1 1 1 1 2 6 8
6	0 0 1 1 2 4 4 4 4 4 8 8 9
7	0 5 5 5 5 7
8	3 4 4 5 6 6 6 7 8 9
9	0 1 2 2 2 2 5 5 6 8 9 9
10	2 2 2 5 7

Вихідні дані в рядку:
102, 102, 102, 105, 107

Також шукаємо ключ, це буде значення першого елемента, після якого кожному наступному елементу в 1 стовпчику відповідає масив значень у другому. Для наведеного вище прикладу ключем буде значення $0|5 = 05$

5. Для побудови коробкового графіка нам потрібно застосувати знайдені значення у минулих завданнях, а саме:
- Медіану
 - Перший та третій квартилі
 - Верхня та нижня границя (мінімальне та максимальне значення вибірки)

Ось як буде зображатись графік:



(Зауважимо, що для побудови графіків використовується бібліотека `ruplot`, де за допомогою функції `boxplot()` передаються дані і автоматично визначаються персантілі, медіана. За допомогою цього можна перевірити правильність розрахунків у минулих завданнях, порівняти їх точність)

Код алгоритму:

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

def connectTxT(data):    #імпорт файлу та внесення його значень у змінну,
#створення файлу для запису
    NameOfFile = input("Input txt file: ")
    data = np.genfromtxt(NameOfFile, dtype='int')
    data = np.delete(data,0)
    print("Unsorted data: ", data)
    File.write("Unsorted data: " + str(data))
    data = sorted(data)
    print("\nSorted data: ", data)
    File.write("\nSorted data: " + str(data) + "\n\n")
    return data

def getFirstDigit(num):    #визначення першої цифри з числа
    digit = ""
    for i in range(len(str(num))-1):
        digit += str(num)[i]
    return digit

def Pfunc(x, SortedData):    #обчислення персантілів
    index = x * (len(SortedData) + 1) - 1
    Percentile = SortedData[int(index)] + (index % int(index)) *
(SortedData[int(index) + 1] - SortedData[int(index)])
    return Percentile

def StandartDeviation(data, midleX):    #стандартне відхилення
    totalSum = 0
    for i in range(len(data)):
        totalSum += (data[i] - midleX)**2
    return np.sqrt(totalSum/(len(data)-1))

def AverageDeviation(data, midleX):    #середнє відхилення
    totalSum = 0
    for i in range(len(data)):
        totalSum += abs(data[i] - midleX)
    return totalSum/(len(data))

def boxdiagram(data):    #коробкова діаграма
    plt.figure(figsize=(10, 7))
    plt.boxplot(data)
    plt.grid()
    plt.show()
```

```

def lineal(data, midleX):    #обчислення оцінок за допомогою лінійних рівнянь
    result = []
    a = np.array([
        [100*1, 1, ],
        [1*midleX, 1, ]
    ])
    b = np.array([100, 95])
    x = solve(a, b)
    print("a = " + str(x[0]) + "\nb = " + str(x[1]))
    File.write("a = " + str(x[0]) + "\nb = " + str(x[1]))
    for i in range(len(data)):
        result.append(x[0]*data[i]+x[1])
    print("Result marks: " + str(result) + "\n\n")
    File.write("\nResult marks: " + str(result) + "\n\n")

Data = connectTxT(Data)
minimal = min(Data)
maximal = max(Data)
Q1 = Pfunc(1 / 4, Data)
Q3 = Pfunc(3 / 4, Data)
P90 = Pfunc(0.9, Data)
sum = 0
for i in range(len(Data)):
    sum += Data[i]
midleX = sum / len(Data)
print("\n\nMediana = ", midleX)
print("\nLower(first) quartile = " + str(Q1))
File.write("Lower(first) quartile = " + str(Q1))
print("Upper(third) quartile = " + str(Q3))
File.write("\nUpper(third) quartile = " + str(Q3))
print("\n90-th percentile = " + str(P90))
File.write("\n\n90-th percentile = " + str(P90))
print("\nStandart Deviation = " + str(StandartDeviation(Data, midleX)))
File.write("\n\nStandart Deviation = " + str(StandartDeviation(Data, midleX)))
print("Average Deviation = " + str(AverageDeviation(Data, midleX)) + "\n")
File.write("\nAverage Deviation = " + str(AverageDeviation(Data, midleX)) +
"\n\n")
lineal(Data, midleX)

print("Stem-Leaf Diagram")
File.write("\nStem-Leaf Diagram")
def CreateTable(data):    # діаграма "стовбур-листя"
    strnumbers = ""
    leaves = []
    key = 0
    for i in range(len(data)):
        strnumbers += str(data[i] % 10) + " "
        FDigit = int(getFirstDigit(data[i]))
        j = i
        if (i == len(data)-1):
            leaves.append(strnumbers)
            print(getFirstDigit(data[i]) + " \t| \t" + strnumbers)
            File.write("\n" + getFirstDigit(data[i]) + " \t| \t" + strnumbers)
            break
        elif (getFirstDigit(data[i]) != getFirstDigit(data[i+1])):
            leaves.append(strnumbers)
            print(getFirstDigit(data[i]) + " \t| \t" + strnumbers)
            File.write("\n" + getFirstDigit(data[i]) + " \t| \t" + strnumbers)
            strnumbers = ""
            FDigit += 1
        while (FDigit != int(getFirstDigit(data[i+1]))):
            print(str(FDigit) + " \t| \t" + strnumbers)
            File.write("\n" + str(FDigit) + " \t| \t" + strnumbers)
            FDigit += 1

```

```

while (j != len(data) and key == 0):
    if (int(getFirstDigit(data[i])) == int(getFirstDigit(data[j]))-1):
        key = data[i]
        break
    else:
        j += 1
print("\nKey: " + str(key))
File.write("\n\nKey: " + str(key))

CreateTable(Data)
boxdiagram(Data)

```

Результат на консолі та випробування алгоритму

Результат для 10 елементів:

```

Input txt file: input_10.txt
Unsorted data: [ 40  65  62  70 100  90  66  70  95  84]

Sorted data: [40, 62, 65, 66, 70, 70, 84, 90, 95, 100]

Mediana = 74.2

Lower(first) quartile = 64.25
Upper(third) quartile = 91.25

90-th percentile = 99.5

Standart Deviation = 18.103406677565783
Average Deviation = 14.440000000000001

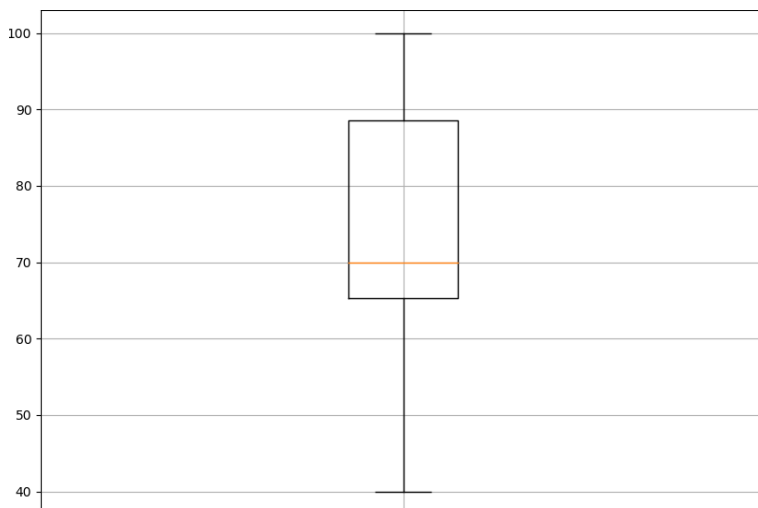
a = 0.19379844961240322
b = 80.62015503875968
Result marks: [88.3720930232558, 92.63565891472868, 93.21705426356588, ...

Stem-Leaf Diagram
4 | 0
5 |
6 | 2 5 6
7 | 0 0
8 | 4
9 | 0 5
10 | 0

Key: 62

```

Графік



Також результат для 100 елементів:

```
Input txt file: input_100.txt
Unsorted 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, ...

Mediana = 57.96

Lower(first) quartile = 36.5
Upper(third) quartile = 78.75

90-th percentile = 91.9

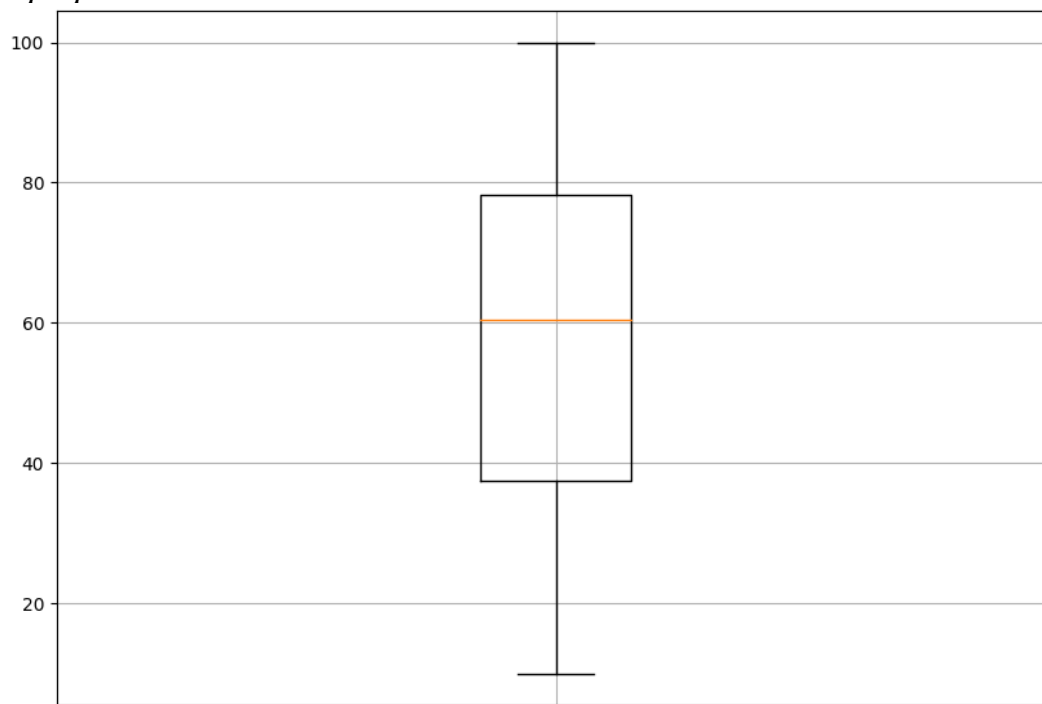
Standart Deviation = 25.280615003387382
Average Deviation = 21.243199999999999

a = 0.1189343482397716
b = 88.10656517602284
Result marks: [89.29590865842056, 89.5337773549001, 89.77164605137965, 89.77164605137965, ...

Stem-Leaf Diagram
1 | 0 2 4 4 5 6 6 9 9
2 | 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

Key: 10
```

Графік:



Висновок:

Під час виконання лабораторної роботи ми навчились використовувати набуті знання про лінійні перетворення та зображення даних. Опанували навички знаходження кватилів, персантилів, стандартного та середнього відхилення. Також навчились будувати коробкові графіки та діаграму «стовбур-листя».