|  |  |  |  |
| --- | --- | --- | --- |
| 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:** | Шекера Олександр  Валерійович | **Перевірила**: | Вечерковська Анастасія Сергіївна |
| Група | ІПЗ-22 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

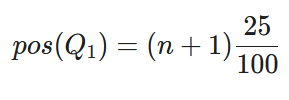
**Тема:** Лінійне перетворення та Графічне зображення даних

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

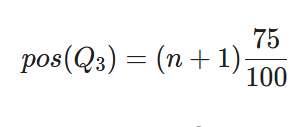
Завдання:  
1. Знайдіть , ,   
2. Знайдіть середнє та стандартне відхилення цих оцінок.  
3. Через незадоволення низькими оцінками викладач вирішив використати  
шкалу форми y = ax + b, щоб відредагувати оцінки. Він хотів, щоб середнє  
значення масштабних оцінок становило 95, а оцінка 100, щоб залишалася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

**Математична модель:**

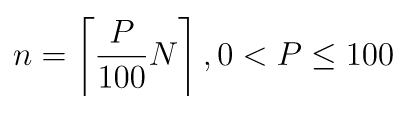
Формула для :



Формула для :



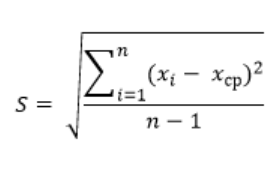
Формула розрахунку порядкового номера n для заданого процентиля *P*:



Формула пошуку середнього квадратичного відхилення:



Формула пошуку стандартного відхилення:



Для зміни середнього значення за лінійною трансформацією використовувалась  
формула:  
𝑦 = 𝑎𝑥 + 𝑏,  
Де : 100 = 100𝑎 + 𝑏  
𝑁𝑒𝑤 𝐴𝑣𝑒𝑟𝑎𝑔𝑒 = 𝑀𝑒𝑎𝑛 + 𝑏

Для побудови коробкового графіку використовувалась бібліотека matplotlib  
мови програмування python.

**Псевдокод алгоритму**

Пошук картелю:

**START:**

**INIT QuartilePos = 0**

**IF pos equals 1**

**QuartilePos = (List.Size + 1) \* (25/100)**

**RETURN List[QuartilePos]**

**ENDIF**

**ELSE IF pos equals 3**

**QuartilePos = (List.Size + 1) \* (75/100)**

**RETURN List[QuartilePos]**

**ENDIF**

**END**

Пошук порядкового номера n для заданного процентиля :

**START**

**N = ((Percent \* List.Size) / 100)**

**RETURN List[N]**

**END**

Пошук дисперсії та середнього відхилення:

**START**

**INIT Variation = 0**

**INIT Avarage = Avarage()**

**INIT Tmp[]**

**FOR from j = 0 to List.Size**

**Tmp[j] = List[j]**

**ENDFOR**

**FOR from i = 0 to Tmp.Size**

**Tmp[i] = (Tmp[i] - Avarage) ^ 2**

**ENDFOR**

**FOR from i = 0 to I Tmp.Size**

**Variation += Tmp[i]**

**ENDFOR**

**Variation = Variation / Tmp.Size**

**INIT Difference = Sqrt(Variation)**

**RETURN Difference**

**END**

Пошук стандартного відхилення:

**START**

**INIT Variation = 0**

**INIT Avarage = Avarage()**

**INIT Tmp[]**

**FOR from j = 0 to List.Size**

**Tmp[j] = List[j]**

**ENDFOR**

**FOR from i = 0 to Tmp.Size**

**Tmp[i] = (Tmp[i] - Avarage) ^ 2**

**ENDFOR**

**FOR from i = 0 to I Tmp.Size**

**Variation += Tmp[i]**

**ENDFOR**

**Variation = Variation / (Tmp.Size - 1)**

**INIT Difference = Sqrt(Variation)**

**RETURN Difference**

**END**

Задача про студентів:

**START**

**Avarage = Avarage()**

**INIT Num = 95**

**A = (100 - Num) / (100 - Avarage)**

**B = Num – Avarage \* A**

**RETURN A, B**

**END**

**Код алгоритму:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace Program

{

class FileWork

{

private string fileName;

private string path = @"C:\Users\oleks\Desktop\кну\теорвер\lab2\task\_02\_data\input\_100.txt";

private List<int> list = new List<int>();

public FileWork()

{

}

public FileWork(string fileName)

{

this.fileName = fileName;

}

public void Read()

{

StreamReader streamReader = File.OpenText(path);

string input;

while ((input = streamReader.ReadLine()) != null)

{

list.Add(Convert.ToInt32(input));

}

streamReader.Close();

}

public void Write()

{

var stream = new FileStream(path + "test1.txt", FileMode.Create, FileAccess.Write);

var writer = new StreamWriter(stream, Encoding.GetEncoding(1251));

list.Sort();

foreach (int num in list)

{

Console.Write(num + " ");

}

writer.WriteLine("\tRank\t\tX");

for (int i = 0; i < list.Count; i++)

{

writer.WriteLine($"\t{i + 1}\t\t{list[i]}");

}

writer.WriteLine($"\nQ1: {Quartile(1)}\tQ3: {Quartile(3)}\tPercentile(90): {Percentile(90)}");

writer.WriteLine($"\nMean Deviation: {MeanDeviation()}\tStandart Deviation: {StandartDeviation()}");

double[,] vs = Task3(95);

writer.WriteLine($"\nAvarage: {Avarage()}\ty = {vs[0,0]}x + {vs[1,1]}");

writer.WriteLine($"Stem\t|\tLeaf");

for(int i = 0; i < list.Count; i++)

{

if (i != 0)

{

if (list[i] /10 == list[i - 1] /10)

{

if (list[i] > 99)

{

writer.Write($", {list[i] % 100}");

}

else

{

writer.Write($", {list[i] % 10}");

}

}

else

{

if (list[i] > 99)

{

writer.Write($"\n{list[i] / 10}\t|\t{list[i] % 100}");

}

else

{

writer.Write($"\n{list[i] / 10}\t|\t{list[i] % 10}");

}

}

}

else

{

if (list[i] > 99)

{

writer.Write($"{list[i] / 10}\t|\t{list[i] % 100}");

}

else

{

writer.Write($"{list[i] / 10}\t|\t{list[i] % 10}");

}

}

}

writer.Close();

}

private int Quartile(int pos)

{

double quartile\_pos = 0;

if (pos == 1)

{

quartile\_pos = (list.Count + 1) \* (25.0 / 100.0);

return list[Convert.ToInt32(quartile\_pos)];

}

else if (pos == 3)

{

quartile\_pos = (list.Count + 1) \* (75.0 / 100.0);

return list[Convert.ToInt32(quartile\_pos)];

}

else

{

throw new IndexOutOfRangeException();

}

}

private int Percentile(int percent)

{

int n = ((percent \* list.Count)/100);

return list[n];

}

public double Avarage()

{

double sum = 0;

foreach (int elem in list)

{

sum += elem;

}

return sum / list.Count;

}

private double MeanDeviation()

{

double variation = 0;

double avarage = Avarage();

double[] tmp = new double[list.Count];

for (int j = 0; j < list.Count; j++)

{

tmp[j] = list[j];

}

for (int i = 0; i < tmp.Length; i++)

{

tmp[i] = Math.Pow((double)(tmp[i] - avarage), 2);

variation += tmp[i];

}

variation = variation / tmp.Length;

double difference = Math.Sqrt(variation);

return difference;

}

private double StandartDeviation()

{

double variation = 0;

double avarage = Avarage();

double[] tmp = new double[list.Count];

for (int j = 0; j < list.Count; j++)

{

tmp[j] = list[j];

}

for (int i = 0; i < tmp.Length; i++)

{

tmp[i] = Math.Pow((double)(tmp[i] - avarage), 2);

variation += tmp[i];

}

variation = variation / (tmp.Length - 1);

double difference = Math.Sqrt(variation);

return difference;

}

private double[,] Task3(int num)

{

double[,] result = new double[2,2];

double avarage = Avarage();

double a = (100 - num) / (100 - avarage);

double b = num - avarage \* a;

result[0, 0] = a;

result[1, 1] = b;

return result;

}

}

}

**Код побудови графіка:**

import math

import matplotlib.pyplot as plt

import numpy as np

f = open('C:/Users/oleks/Desktop/кну/теорвер/лаб1/task\_01\_data/input\_100.txt', 'r')

values = []

for i in f:

    values.append(int(i))

values.sort()

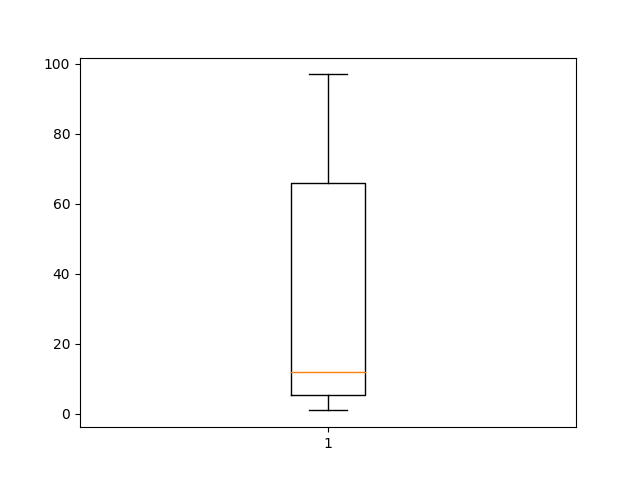
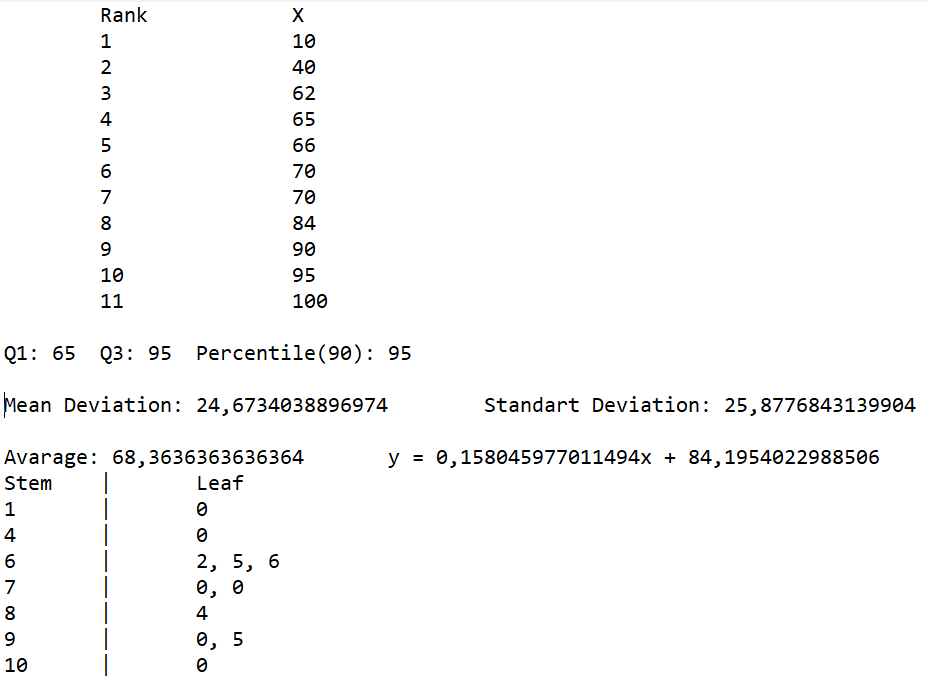
plt.boxplot(values)

plt.savefig('boxplot')

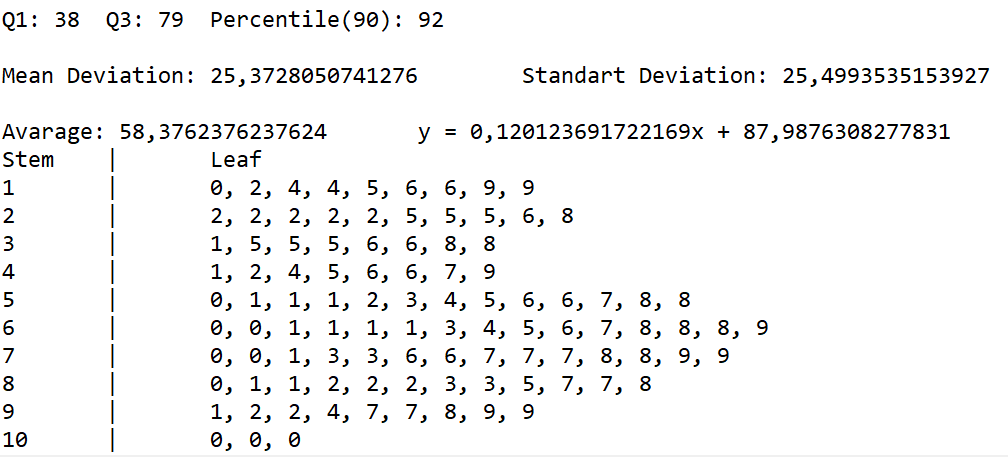
plt.show()

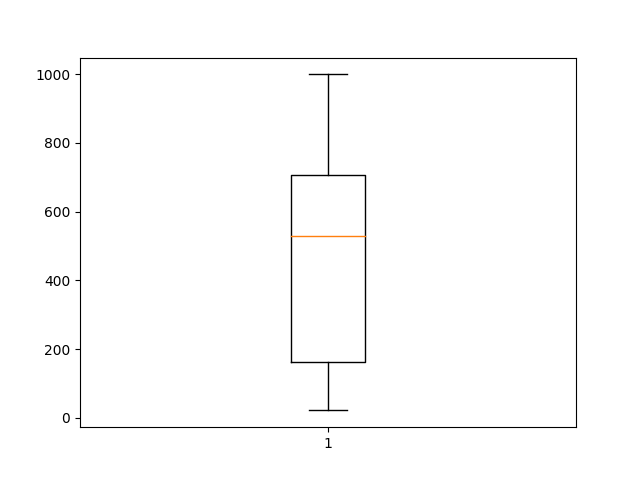
**Випробування алгоритму:**

Для перевірки запустимо програму на файлі, з 11 значеннями:



Результат роботи програми при вхідних даних розміром 100:





**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 квартиль,  
90-ий процентиль, змінює розподіл з використанням формули y = ax + b,  
відображає дані за допомогою діаграми «стовбур-листя» та коробкового  
графіку.