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:	Коваленко Владислав Олександрович	**Перевірила**:	Марцафей А. С.
Група	ІПЗ-22(2)	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Назва роботи: Лінійне перетворення та Графічне зображення даних

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

Завдання №1:

Постановка задачі:
Знайдіть Q_1, Q_3 та P90

Побудова математичної моделі:
Для розв'язання даної задачі нам потрібно порахувати переписати дані з нашого файлу в масив.Далі шукаємо індекс першого Квартиля за допомогою формули index = 1/4*(n+1) n- кількість елементів в послідовності.Далі шукаємо Перший квартиль за формулою Q_1= mas[index] + 1/4(mas[index +1]-mas[index-1]).Третій квартиль знаходиться ідентично ,але замість 1/4 ставимо 3/4. P_{90} рахується за тою ж формулою але замість 1/4 ставимо 90/100,оскільки перший квартиль являється 25-тим персентилем.

Псевдокод алгориму:

```
def Task1():
    print("Завдання 1")
    filename = input("Введіть назву файлу: ")

    data = open(filename).read().splitlines()
    del data[0]
    mas = [int(x) for x in data]
    mas.sort()
    outputfile = open("OutputFile.txt","w")
    indexQ1 = (1/4)*(len(mas) +1)
    i,d = divmod(indexQ1,1)
    outputfile.writelines("==========Task #1==============")
    outputfile.writelines("\nQ1 = "+str(mas[int(i-1)])+ " + "+"1/4*(" +str(mas[int(i)]) +" - "+str(mas[int(i-1)])+")")
    Q1 = mas[int(i-1)] + d*(mas[int(i)] - mas[int(i-1)])
    outputfile.writelines("\nQ1 = "+str(Q1))
    indexQ3 = (3/4)*(len(mas) +1)
    i,d = divmod(indexQ3,1)
    outputfile.writelines("\nQ3 = "+str(mas[int(i-1)])+ " + "+"3/4*(" +str(mas[int(i)]) +" - "+str(mas[int(i-1)])+")")
    Q3 = mas[int(i-1)] + d*(mas[int(i)] - mas[int(i-1)])
    outputfile.writelines("\nQ3 = "+str(Q3))
    indexP90 = round((90/100)*(len(mas) ))
    outputfile.writelines("\nP90 = "+str(mas[int(indexP90-1)])+ " + "+"90/100*(" +str(mas[int(indexP90)]) +" - "+str(mas[int(indexP90-1)])+")")
    P90 = mas[indexP90-1] + 90/100*(mas[indexP90-1] - mas[indexP90-1])
    outputfile.writelines("\nP90 = "+str(P90))
```

Випробування алгоритму:

```
==========Task #1==============
Q1 = 62 + 1/4*(65 - 62)
Q1 = 64.25
Q3 = 90 + 3/4*(95 - 90)
Q3 = 91.25
P90 = 95 + 90/100*(100 - 95)
P90 = 95.0
```

Завдання 2:

Постановка задачі:
Знайдіть середнє та стандартне відхилення цих оцінок.
Побудова математичної моделі:
Для розв'язання даної задачі нам потрібно порахувати переписати дані з нашого файлу в масив.Далі шукаємо середнє арифметичне за допомогою циклу.І рахуємо середнє та стандартне відхилення цих оцінок за допомогою

$$\sigma = \frac{\sum_i^N |X_i - \bar{X}|}{N}$$

формули .А результат записуємо в файл

Псевдокод алгориму:

```
def Task2():
    print("Завдання 2")
    outputfile = open("OutputFile.txt","a")
    outputfile.writelines("\n==========Task #2==========")
    filename = input("Введіть назву файлу: ")
    avarage = 0
    data = open(filename).read().splitlines()
    del data[0]
    mas = [int(x) for x in data]
    for i in range(len(mas)):
        avarage += mas[i]
    avarage = avarage/len(mas)
    AverageDeviation = 0
    for i in range(len(mas)):
        AverageDeviation += math.fabs(mas[i]-avarage)
    AverageDeviation /= len(mas)
    outputfile.writelines("\nСереднє та стандратне відхилення дорівнює - "+str(AverageDeviation))

def Task3():
```

Випробування алгоритму:

```
==========Task #2==========
Середнє та стандратне відхилення дорівнює - 14.440000000000003
```

Завдання 3:

Постановка задачі:
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.

Побудова математичної моделі:
Для розв'язання даної задачі нам потрібно розв'язати систему рівнянь

```
{ 100 = 100*a + b
{ 95 = 74.2*a + b
```

Це можна зробити за допомогою функції linalg.solve() бібліотеки NumPy.Розв'язавши дану систему рівнянь перемножуємо кожний елемент в масиві на ці коефіцієнти і переписуємо нашу послідовність і перераховуємо середнє арифметичне.І в кінці записуємо все в файл

Псевдокод алгоритму:

```
def Task3():
    print("Завдання 3")
    outputfile = open("OutputFile.txt","a")
    outputfile.writelines("\n==========Task #3==========")
    filename = input("Введіть назву файлу: ")
    avarage = 0
    data = open(filename).read().splitlines()
    del data[0]
    mas = [int(x) for x in data]
    for i in range(len(mas)):
        avarage += mas[i]
    avarage = avarage/len(mas)
    outputfile.writelines("\nВхідний масив: "+str(mas))
    outputfile.writelines("\nСереднє арифметичне: "+str(avarage))
    M1 = np.array([[100.,1.],[avarage,1.]])
    V1 = np.array([100.,95.])
    outputfile.writelines("\n{ 100 = 100*a + b")
    outputfile.writelines("\n{ 95 = "+str(avarage)+"*a" +" + b")
    Res = np.linalg.solve(M1, V1)
    a = Res[0]
    b = Res[1]
    outputfile.writelines("\n a = "+str(a)+", b = "+str(b))
    avarage = 0;
    for i in range(len(mas)):
        mas[i] = mas[i]*a +b
    for i in range(len(mas)):
        avarage += mas[i]
    avarage = avarage/len(mas)
    outputfile.writelines("\nВихідний масив: "+str(mas))
    outputfile.writelines("\nСереднє арифметичне: "+str(avarage))
```

Випробування алгоритму:

```
Вхідний масив: [40, 65, 62, 70, 100, 90, 66, 70, 95, 84]
Середнє арифметичне: 74.2
{ 100 = 100*a + b
{ 95 = 74.2*a + b
 a = 0.19379844961240308, b = 80.62015503875969
Вихідний масив: [88.37209302325581, 93.2170542635659, 92.63565891472868, 94.18604651162791,
100.0, 98.06201550387597, 93.4108527131783, 94.18604651162791, 99.03100775193798, 96.89922480620154]
Середнє арифметичне: 95.0
```

Завдання 4:

Постановка задачі:
Показати дані за допомогою діаграми "стовбур – листя".

Побудова математичної моделі:
Для побудови діаграми Стовбур-Листя нам потрібно відсортувати масив за зростянням.Далі використовуючи цикл шукаємо відповідні елементи відповідні до індексу і записуємо все в текстовий файл.І в кінці записуємо ключ до нашої діаграми

Псевдокод алгоритму:

```
def Task4():
    print("Завдання 4")
    outputfile = open("OutputFile.txt","a")
    outputfile.writelines("\n==========Task #4==============")
    filename = input("Введіть назву файлу: ")
    avarage = 0
    data = open(filename).read().splitlines()
    del data[0]
    mas = [int(x) for x in data]
    mas.sort()
    outputfile.writelines("\n==========Діаграма Стовбур-Дерево==============")
    for i in range(len(mas)+1):
        outputfile.write("\n"+str(i)+") ")
        for j in range(len(mas)):
            if mas[j] < ((i*10)+10) and mas[j] >= i*10:
                outputfile.write("\t"+(str(mas[j]-i*10)))
    outputfile.writelines("\nКлюч 4|0 = 40")
```

Випробування алгоритму:

```
==========Task #4==============
==========Діаграма Стовбур-Дерево==============
0)
1)
2)
3)
4)      0
5)
6)      2       5       6
7)      0       0
8)      4
9)      0       5
10)     0
Ключ 4|0 = 40
```

Завдання 5:

Постановка задачі:
Відобразити дані за допомогою коробкового графіка.

Побудова математичної моделі:
Для побудови коробкового графіка нам потрібно переписати елементи з файлу в масив і відсортувати його.Далі за допомогою функції boxplot() бібліотеки MatPlotLib.Pyplot будуємо графік і зберігаємо його в файлі.

Псевдокод алгоритму:

```
def Task5():
    print("Завдання 5")
    outputfile = open("OutputFile.txt","a")
    filename = input("Введіть назву файлу: ")
    avarage = 0
    data = open(filename).read().splitlines()
    del data[0]
    mas = [int(x) for x in data]
    mas.sort()
    plt.title("Завдання 5 (Коробковий графік)")
    plt.boxplot(mas)
    plt.savefig("Task5")
    plt.show()
```

Випробування алгоритму:

Висновок:
Виконавши цю лабораторну роботу, я навчився використовувати на практиці набуті знання про лінійні перетворення та графічне зображення даних.