

Київський національний університет ім. Т.Г.Шевченка
кафедра Програмних систем та Технологій

Звіт
до лабораторної роботи № 2
варіант №11
з дисципліни
«Об'єктно-орієнтоване конструювання програм»
Студента групи ІПЗ-24 групи
Липник Артем Вадимович

Лабораторна робота 2

Лінійне перетворення та Графічне зображення даних

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

Постановка задачі

1. Знайдіть Q1, Q3, та P90
2. Знайдіть середнє та стандартне відхилення цих оцінок.
3. Через незадоволення низькими оцінками викладач вирішив використати шкалу форми $y = ax + b$, щоб відрегулювати 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.

Псевдокод алгоритму

```
import math
import numpy as np
import matplotlib.pyplot as plt

data = []
countData = 0
f = open("result.txt", "w+")

def setData(fileName, data):
    global countData
    file = open(fileName, 'r')
    for line in file:
        data.append(int(line.strip()))

    countData = data.pop(0)

def getFirstDigit(num):
    gg = ""
    for i in range(len(str(num))-1):
        gg += str(num)[i]
```

```

return gg

def QPFind(x):
    global data
    sortData = data.copy()
    sortData.sort()
    index = x * (len(sortData) + 1) - 1
    Percentile = sortData[int(index)] + (index % int(index)) * \
        (sortData[int(index) + 1] - sortData[int(index)])
    return Percentile

def StandartDeviation(data):
    sum = 0
    resSum = 0
    for i in data:
        sum += i

    avg = sum / len(data)

    for i in data:
        resSum += (i - avg)**2
    return np.sqrt(resSum/(len(data)-1))

def AverageDeviation(data):
    sum = 0
    totalSum = 0
    for i in data:
        sum += i

    avg = sum / len(data)

    for i in data:
        totalSum += abs(i - avg)

    return totalSum/(len(data))

def boxDiagram(data):
    plt.figure(figsize=(10, 7))
    plt.boxplot(data)

```

```

plt.grid()
plt.show()

a = (25/129)
b = (10400/129)

def Task2():
    for i in data:
        y = a * i + b
        print(i, y)

setData('./input_10.txt', data)
Q1 = QPFind(1/4)
Q3 = QPFind(3 / 4)
P90 = QPFind(0.9)
print("\n\nQ1 = ", Q1)
f.write("Q1 = " + str(Q1))
print("\nQ3 = ", Q3)
f.write("\nQ3 = " + str(Q3))
print("\nP90 = ", P90)
f.write("\n\nP90e = " + str(P90))

print("\nStandart Deviation = ", StandartDeviation(data))
f.write("\n\nStandart Deviation = " + str(StandartDeviation(data)))
print("Average Deviation = ", AverageDeviation(data))
print("\n")
f.write("\nAverage Deviation = " + str(AverageDeviation(data)) +
"\n\n")

Task2()

def StemLeafDiagram():
    global data
    sorted = data.copy()
    sorted.sort()
    stems = []
    print(sorted)

```

```

for i in sorted:
    strI = str(i)[0]
    if (i < 100):
        stems.append(int(strI))

    if i == 100:
        stems.append(10)

plt.ylabel('Data')

plt.xlabel('Stems')

plt.stem(stems, sorted)
plt.show()

# StemLeafDiagram()
boxDiagram(data)

```

Випробування алгоритму

Q1 = 64.25

Q3 = 91.25

P90e = 99.5

Standart Deviation = 18.103406677565783

Average Deviation = 14.440000000000003

40 88.37209302325581

65 93.2170542635659

62 92.63565891472868

70 94.18604651162791

100 100.0

90 98.06201550387597

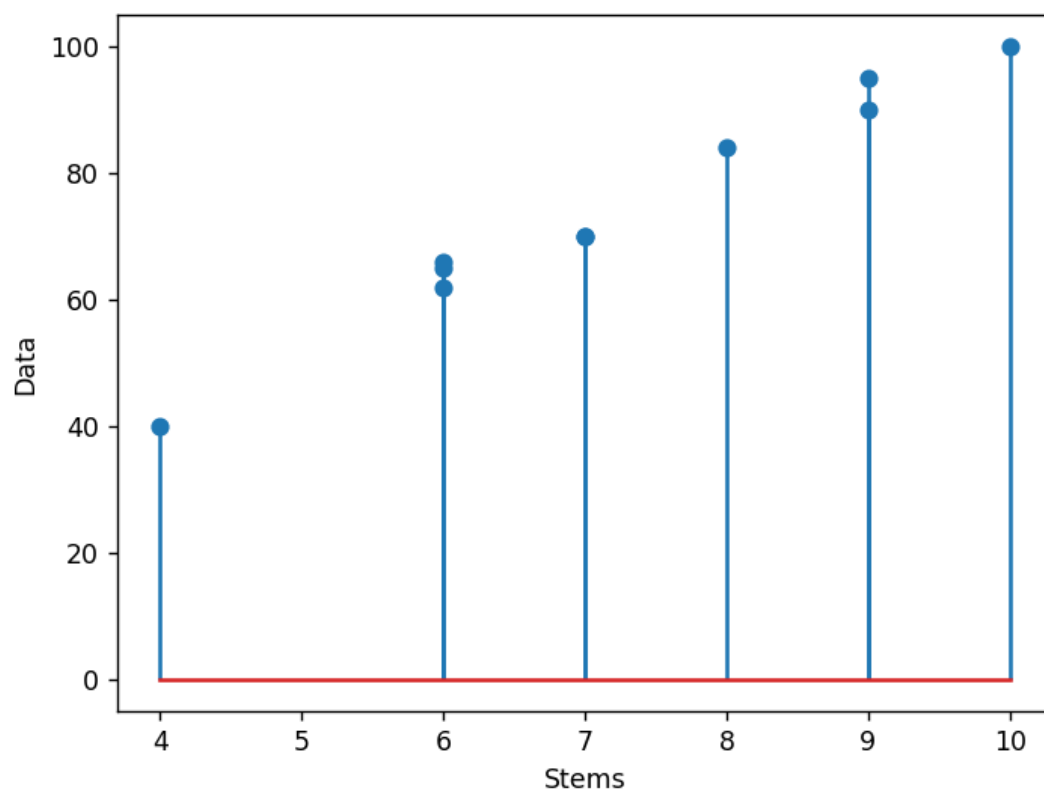
66 93.4108527131783

70 94.18604651162791

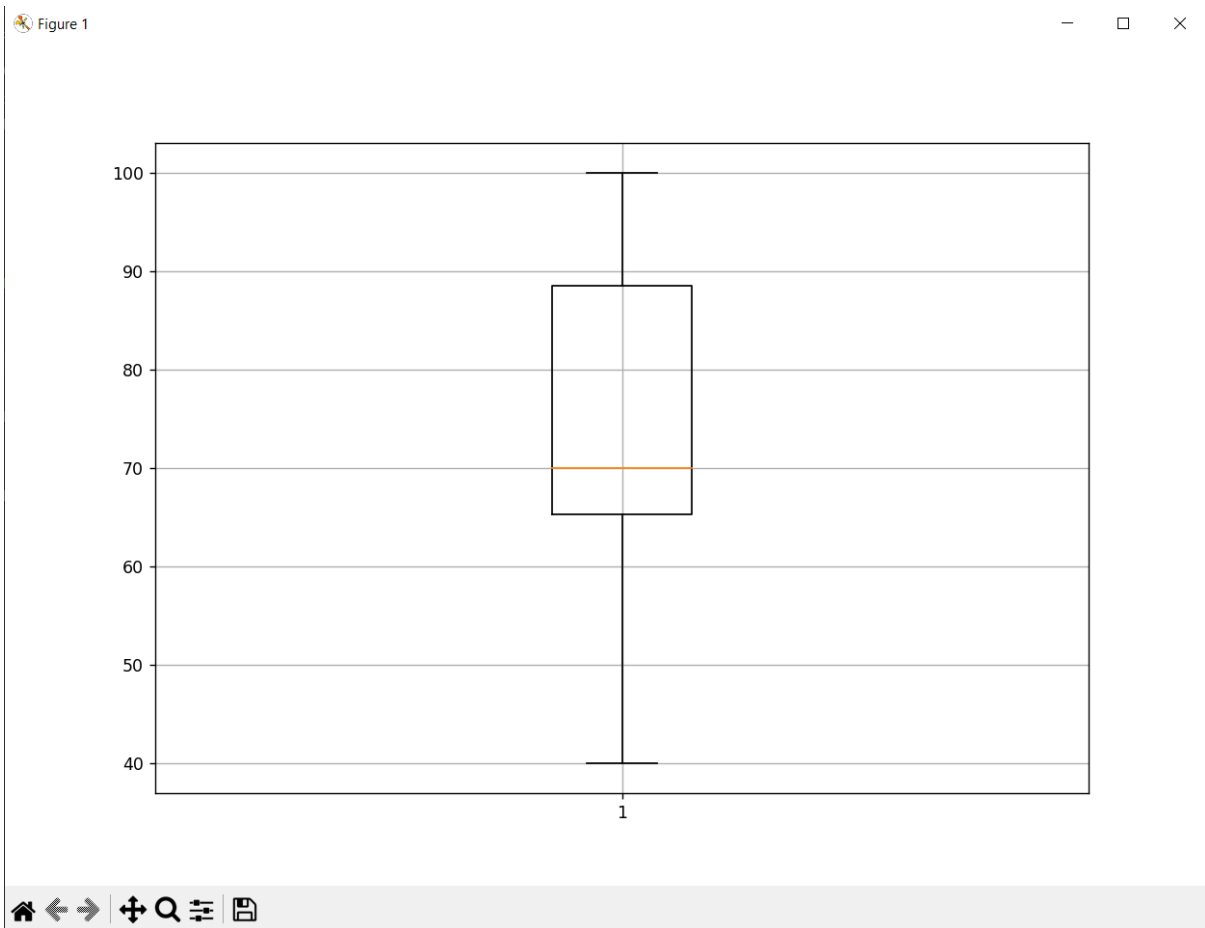
95 99.03100775193799

84 96.89922480620154

Figure 1



x=6.21 y=70.0



Висновок: Проробивши лабораторну роботу я навчився використовувати на практиці набуті знання про лінійні перетворення та графічне зображення даних