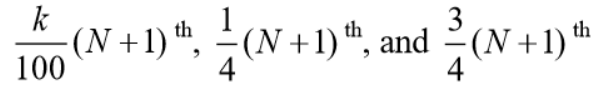
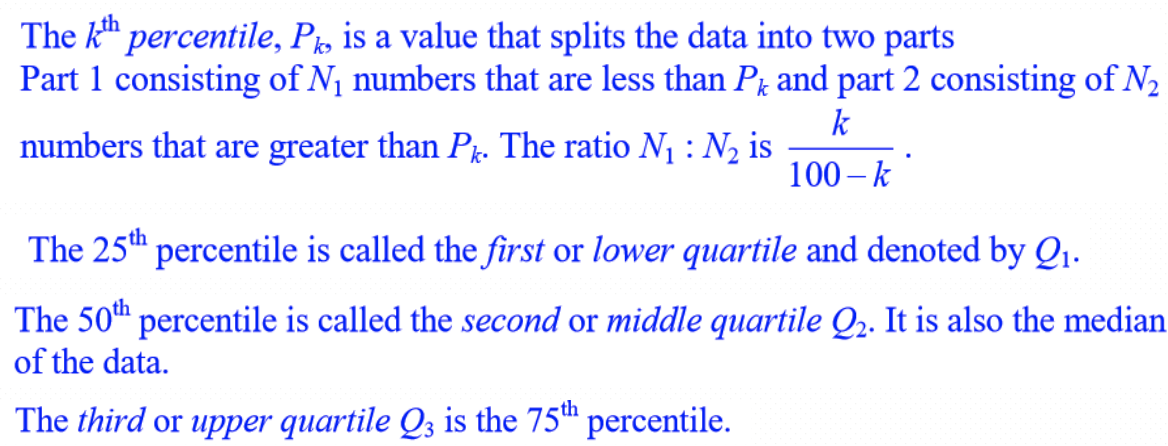
|  |  |  |  |
| --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка  ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**    Дисципліна  **«Ймовірнісні основи програмної інженерії»**  **Лабораторна робота № 2**  **Лінійне перетворення та Графічне зображення даних** | | | |
| **Виконав:** | Бережний В. О. | **Перевірила**: | Вечерковська А. С. |
| Група | ІПЗ-25мс | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

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

**Github посилання:** <https://github.com/Vladyslbr/vpi00lab>

**Хiд роботи**

**Завдання 1** – знайти Q1, Q3, P90



**Код:**

file = loadtxt("input\_10.txt", dtype=int)

file = file[1:]

file = sort(file)

filesSize = file.size

fileVector = file.reshape(-1, 1)

w = arange(1, filesSize + 1, 1)

e = insert(fileVector, 0, w, axis=1)

r = e[..., 1] # working unit

i = e[..., 0]

unique\_values = unique(r)

unique\_values, occurence = unique(r, return\_counts=True)

nf = open('new\_file.txt', 'w')

nf.write('Occurence array:\n'+str(occurence))

q1 = 1/4\*(filesSize + 1)

q3 = 3/4\*(filesSize + 1)

q1 = int(q1)

q3 = int(q3)

q1c1 = i==q1

q1c2 = i==(q1+1)

q1v1 = extract(q1c1, e[..., 1])

q1v2 = extract(q1c2, e[..., 1])

q1r = q1v1 + 0.25\*(q1v2 - q1v1)

q3c1 = i==q3

q3c2 = i==(q3+1)

q3v1 = extract(q3c1, e[..., 1])

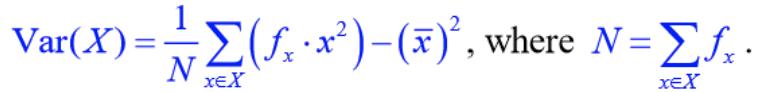
q3v2 = extract(q3c2, e[..., 1])

q3r = q3v1 + 0.75\*(q3v2 - q3v1)

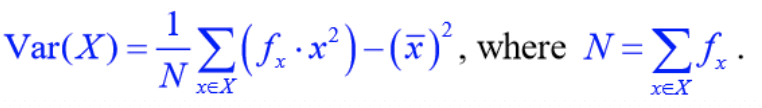
p90 = 90/100 \* (filesSize + 1)

**Завдання 2** – знайти середнє та стандартне відхилення цих оцінок.

Mean square error:



Standart dev:



**Код:**

xMean = sum(unique\_values) / len(unique\_values)

xMean = round(xMean, 4)

S = sqrt((sum(occurence \* square(unique\_values - xMean))) / (sum(occurence) - 1))

S = round(S, 4)

MD = sum(occurence \* absolute(unique\_values - xMean))/sum(occurence)

MD = round(MD, 4)

**Завдання 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.

**Код:**

diffA = array([[xMean, 1], [100, 1]])

diffB = array([[95], [100]])

a, b = linalg.solve(diffA, diffB)

eq = a \* r + b

eq = eq.astype(int)

**Завдання 4** - показати дані за допомогою діаграми "стовбур – листя".

**Код:**

def stemleaf(x):

d = OrderedDict((((str(v)[:-1], ' ')[v < 10], Counter()) for v in sorted(x)))

for s in ((str(v), ' '+str(v))[v < 10] for v in x):

d[s[:-1]][s[-1]] += 1

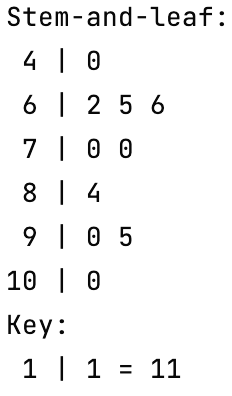
m = max(len(s) for s in d)

for k in d:

write('\n%s%s | %s'%(' '\*(m-len(k)), k, ' '.join(sorted(d[k].elements()))))

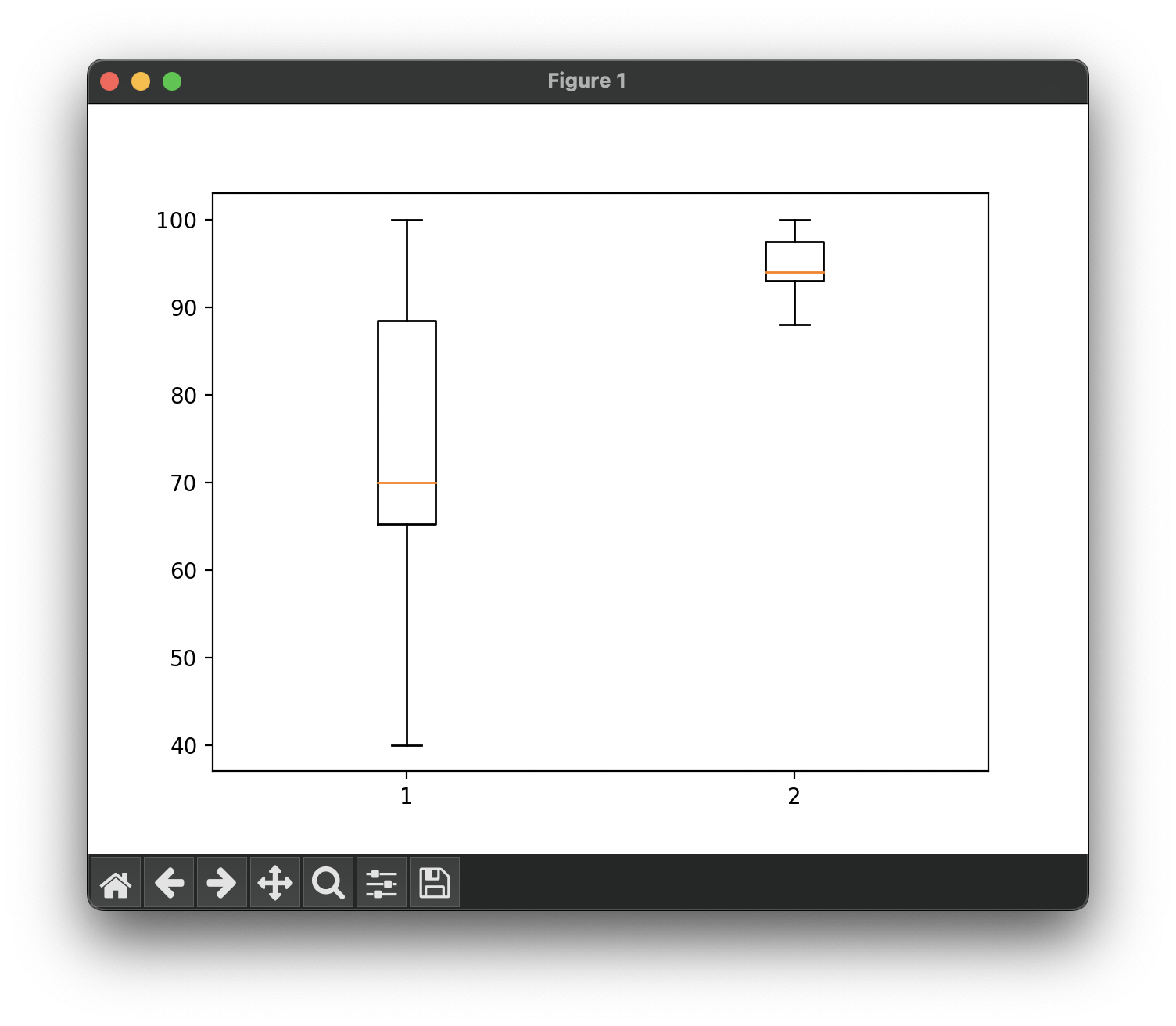
stemleaf(r)

**Результат:**

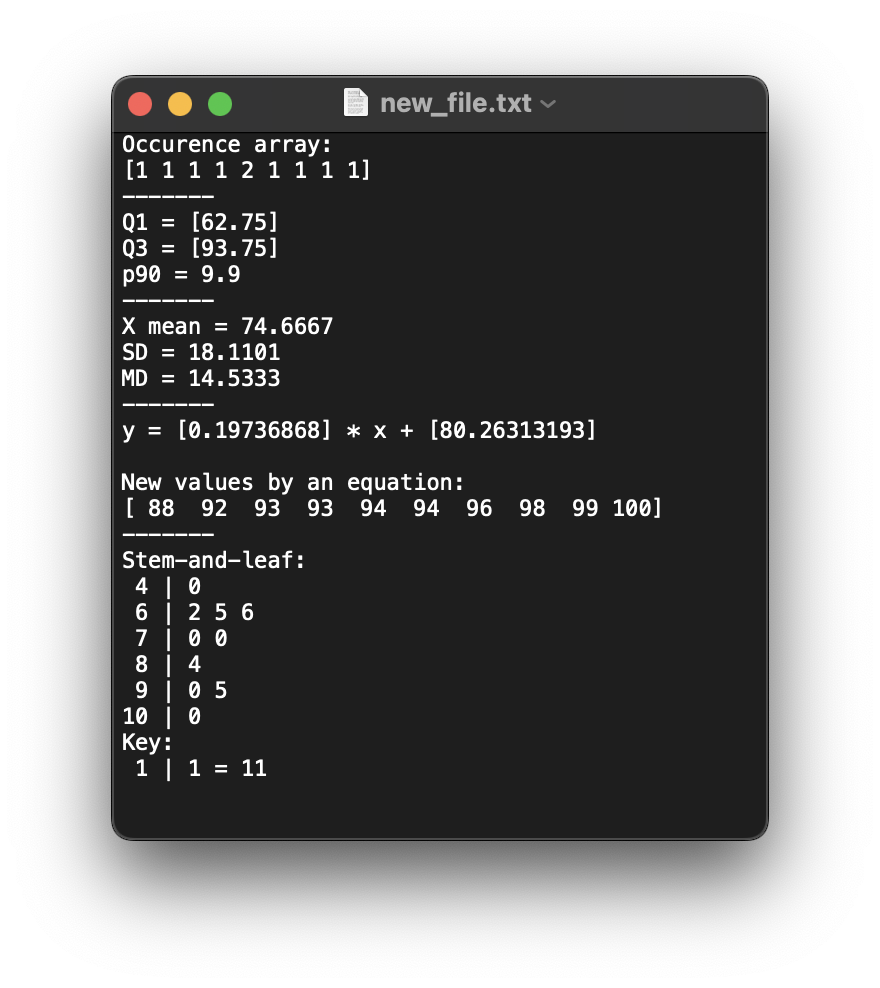


**Завдання 5** - відобразити дані за допомогою коробкового графіка.

Два графiка: побудований по заданому масиву, та вiдредагований по рiвнянню.



**Вихiдний файл з повним результатом:**



**Висновок:** по графiку з завдання 5 можемо сказати, що у вхiдний даних був досить сильний розкид по оцiнках, та як результат застосування формули y = ax + b, що можна побачити на другому коробковому графiку, що розкид суттево зменшився, та усi оцiнки приблизились до показника у 100 балiв.