|  |  |  |  |
| --- | --- | --- | --- |
| 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 № 1**  **Центральні тенденції та міра дисперсії** | | | |
| **Виконав:** | Бережний В. О. | **Перевірила**: | Вечерковська А. С. |
| Група | ІПЗ-25мс | Дата перевірки | 19.10.2022 |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

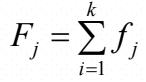
**Мета роботи:** навчитись використовувати на практиці набуті знання про центральні тенденції та міри.

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

**Хiд роботи**

**Завдання 1** Побудувати таблицю частот та сукупних частот для переглянутих фільмів. Визначити фільм, який був переглянутий частіше за інші.

1. Визначаємо частоту переглянутих фiльмiв та сукупну частоту за формулою:



2. Будуємо таблицю частот та сукупних частот, записуємо iх до файлу

3.Знаходимо найбiльш переглянутий фiльм по його частотi та записуємо результат до таблицi

**Код:**

*# Text file data converted to integer data type*

File\_data = np.loadtxt("input\_100.txt", dtype=int)

File\_data = File\_data[1:]

File\_data = np.sort(File\_data) *# sorting values*

lenght = len(File\_data)

unique\_values = np.unique(File\_data)

*# creating indices and occurrence\_count array*

unique\_values, indices\_list = np.unique(File\_data, return\_index=True) *# create index*

unique\_values, occurrence\_count = np.unique(File\_data, return\_counts=True) *# count frequency*

*# cumulative frequency array*

cumulative\_frequency = np.cumsum(occurrence\_count)

*# sum of frequency and cumulative frequency*

occurrence\_count\_sum = np.sum(occurrence\_count)

cumulative\_frequency\_sum = np.sum(cumulative\_frequency)

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

*# recording a table to a file*

pd.set\_option('display.max\_rows', 1000)

df1 = pd.DataFrame({'value':unique\_values, 'freq':occurrence\_count, 'cumFreq':cumulative\_frequency})

nf.write(str(df1))

nf.write('\n\n\nTotal of frequency = '+str(occurrence\_count\_sum))

nf.write('\nTotal of cumulative frequency = '+str(cumulative\_frequency\_sum))

*# the value of most frequent value(s)*

max\_value = occurrence\_count.max()

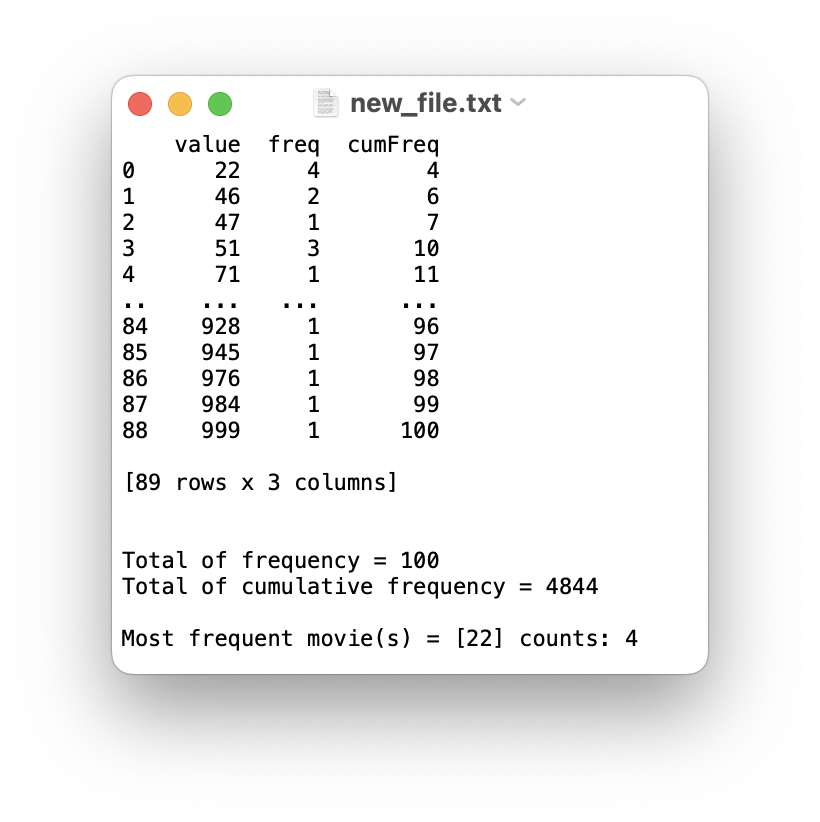
min\_value = occurrence\_count.min()

val\_equalsto\_max\_value = np.nonzero(occurrence\_count == max\_value)

most\_freq\_value = unique\_values[val\_equalsto\_max\_value]

nf.write('\n\nMost frequent movie(s) = '+str(most\_freq\_value)+' counts: '+str(max\_value))

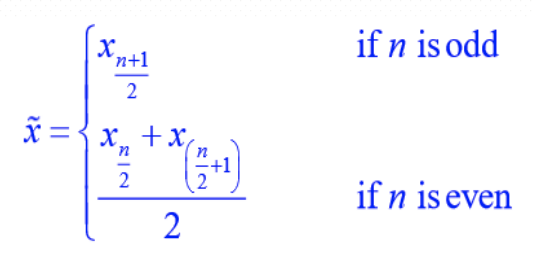
**Результат:**



**Завдання 2** Знайти Моду та Медіану заданої вибірки.

1. Знаходимо медiану з поточноi вибiрки за допомогою формули:

,



результат заносимо до документу.

2. Знаходимо моду та заносимо результат до документу

**Код:**

# finding Me

if (lenght % 2) == 1:

nf.write('\n\nMe(odd) = '+str(File\_data[int(lenght/2)]))

else:

nf.write('\n\nMe(even) = '+str(np.median(File\_data)))

# finding Mo

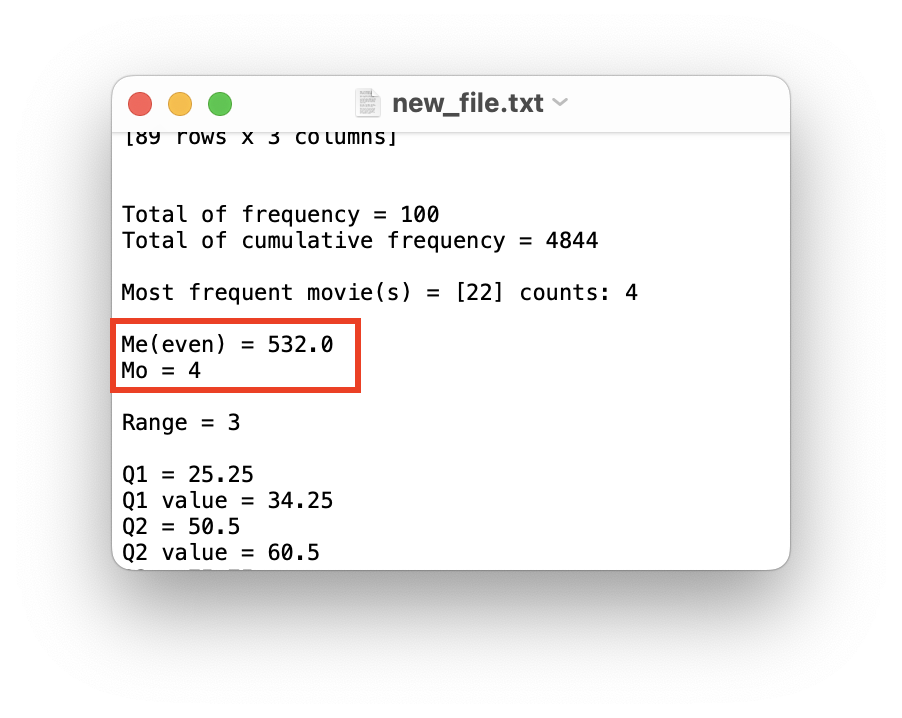
if (max\_value == min\_value): # Mo if no values occurrence

nf.write('\nMo = no value')

else:

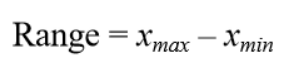
nf.write('\nMo = '+str(max\_value))

**Результат:**



**Завдання 3** Порахувати Дисперсію та Середнє квадратичне відхилення розподілу.

1. Знаходимо Range за формулою:



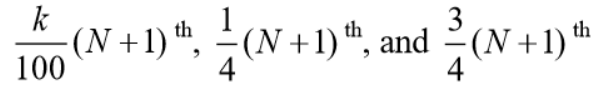
**Код:**

*# finding the Range*

nf.write('\n\nRange = '+str(max\_value - min\_value))

cumulative\_frequency, cum\_indices = np.unique(cumulative\_frequency, return\_index=True)

2. Визначаємо квартиль Q1, Q2, Q3 та знаходимо iх значення за формулами:



**Код:**

*# finding Quartiles' values*

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

nf.write('\n\nQ1 = '+str(q1))

q1 = int(q1)

q11 = cum\_indices[q1]

q12 = cumulative\_frequency[q11 - 1]

q13 = cumulative\_frequency[q11]

q14 = q12 + 1/4 \* (q13 - q12)

nf.write('\nQ1 value = '+str(q14))

q2 = 1/2 \* (lenght + 1)

nf.write('\nQ2 = '+str(q2))

q2 = int(q2)

q21 = cum\_indices[q2]

q22 = cumulative\_frequency[q21 - 1]

q23 = cumulative\_frequency[q21]

q24 = q22 + 1/4 \* (q23 - q22)

nf.write('\nQ2 value = '+str(q24))

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

nf.write('\nQ3 = '+str(q3))

q3 = int(q3)

q31 = cum\_indices[q3]

q32 = cumulative\_frequency[q31 - 1]

q33 = cumulative\_frequency[q31]

q34 = q32 + 1/4 \* (q33 - q32)

nf.write('\nQ3 value = '+str(q34))

3. Знаходимо IQR (inter-quartile range) за формулою:



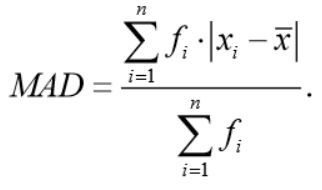
**Код:**

*# finding the inter-quartile range*

iqr = q34 - q14

nf.write('\n\nIQR = '+str(iqr))

4. Знаходимо MAD (mean absolute value) за формулою:



**Код:**

*# finding the mean absolute deviation*

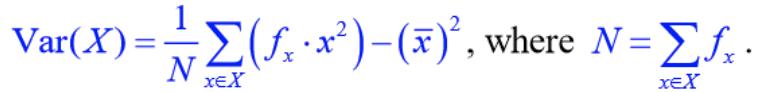
x\_mean = np.sum(cumulative\_frequency) / len(cumulative\_frequency)

mad = np.sum(np.absolute(cumulative\_frequency - x\_mean))/lenght

mad = round(mad, 4)

nf.write('\n\nMAD = '+str(mad))

5. Визначаємо MSE (mean square error) за формулою:



**Код:**

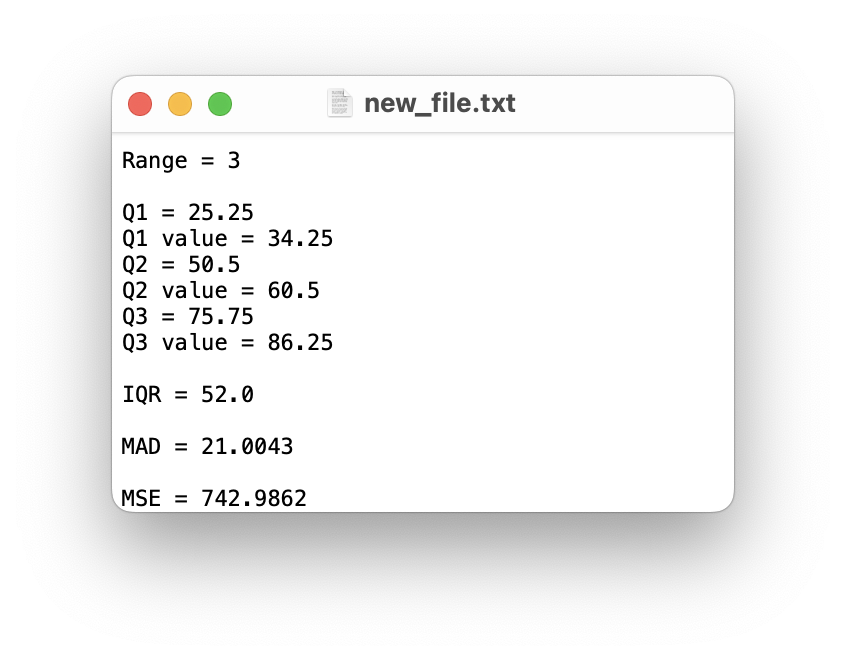
*# finding the mean square error*

mse = 1/len(cumulative\_frequency) \* np.sum(np.square(cumulative\_frequency - x\_mean))

mse = round(mse, 4)

nf.write('\n\nMSE = '+str(mse))

**Результат:**



**Завдання 4-5** Побудувати гістограму частот для даного розподілу та зробити висновок з вигляду гістограми, про закон розподілу.

Будуємо гiстограму частот для вибiрки з файлу input\_100.txt

**Код:**

File\_data = np.loadtxt("input\_100.txt", dtype=**int**)

File\_data = File\_data[1:]

File\_data = np.sort(File\_data) *# sorting values*

unique\_values = np.unique(File\_data)

unique\_values, occurrence\_count = np.unique(File\_data, return\_counts=True)

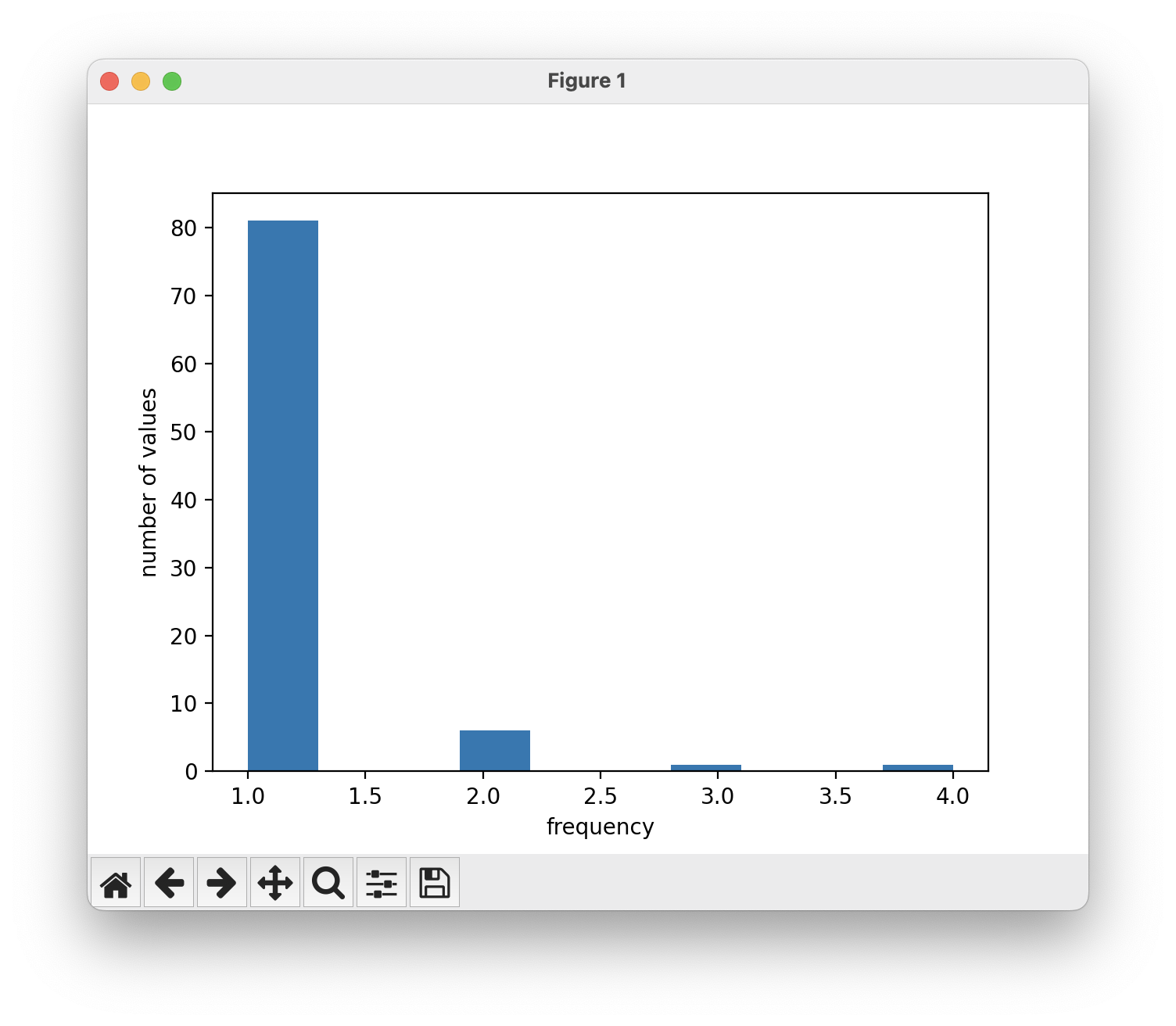
histogram = plt.hist(occurrence\_count)

plt.xlabel('frequency')

plt.ylabel('number of values')

plt.show()

**Результат:**



**Висновок:** по гiстограмi можемо сказати, що розподiл є нерiвномiрним; бiльшiсть значень (80) мають частоту 1; менше десяти значень мають частоту 2, 3, та 4.