|  |  |  |  |
| --- | --- | --- | --- |
| 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:** | Пономаренко Андрій Сергійович | **Перевірила**: | Вечерковська Анастасія Сергіївна |
| Група | ІПЗ-25мс | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

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

**Завдання:**

1. Побудувати таблицю частот та сукупних частот для переглянутих фільмів.

2. Знайти Моду та Медіану заданої вибірки.

3. Порахувати Дисперсію та Середнє квадратичне відхилення розподілу.

4. Побудувати гістограму частот для даного розподілу.

5. Зробити висновок з вигляду гістограми, про закон розподілу.

**Рішення:**

2. Для знаходження медіани ми спочатку превіряєми чи кількість парна(if (len % 2) == 1:) і якщо вона не парна ми просто вивдимо значення в позиціі (n-1)/2+1

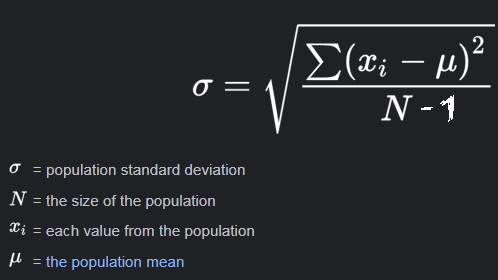
А якщо парна тоді ми знаходимо середнє значень в позиціях n/1 та n/1+1.

Моду ми знаходимо за допомогою функції count\_nonzero що знаходить кількість значень з заданим значенням. По всіх значеннях проходяться та якщо більше значення знайдено попереднє відкидаєтеся

1.Також функцією count\_nonzero знаходимо таблицю частот. Значення та його частота записуються та потім таке саму кількість значень пропускаємо по таблиці щоб не було повторень(список сортовано).

Таблиця кумулятивних частот робится так само але записуются кумулативні частоти замість звичайних.

3.

Стандартне відхилення ми знаходим спочатку знаходячи верхню частину цієї формули у for лупі 

Всі інші операції робляться при записуванні devsum у файл

Варіація знаходится так само але не вводится у квадратиний корінь

4. Гістограма будуєтся за допомогою matplotlib.pyplot.hist()

5.

Дані10:2 щілини;хаотичний розподіл

Дані100:без щілин;відносно нормальний розподіл за виключенням 22

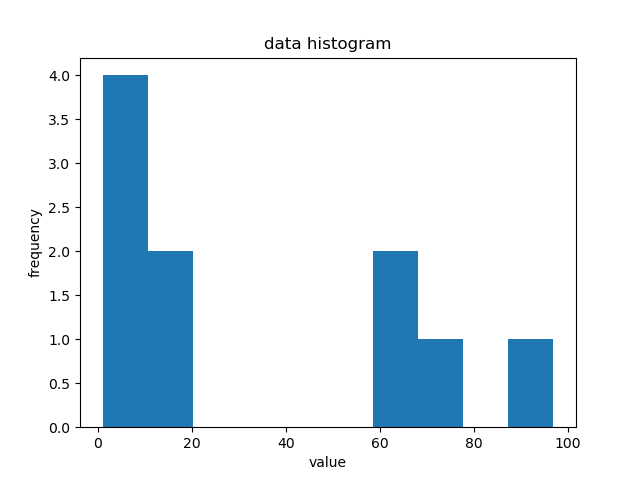
Дані1000:без щілин;рівномірний розподіл

**Код програми:**

#used libraries  
from numpy import \*  
from matplotlib.pyplot import \*  
  
filename = raw\_input()  
f = open(filename,"r")#opens file  
len = int(f.readline())#takes file length from first line  
datarray = []#initialises data array  
for x in range(len):  
 datarray.append(int(f.readline()))  
  
datarray = sort(datarray)  
fig = hist(datarray)#draws histogram  
#writes data properties onto the histogram  
title('data histogram')  
#labels  
xlabel('value')  
ylabel('frequency')  
savefig("output from "+filename+".png")#saves file  
#properties but more manual  
f.close()  
f = open("output from "+filename+".txt","w")  
if (len % 2) == 1:#check odd  
 f.write("Median = "+str(datarray[len/2]))  
else:  
 f.write("Median = "+str((datarray[int(len/2)-1]+datarray[int(len/2)])/2))  
largestCount = 0  
tempfreqarray = datarray  
f.write("\ntable of frequencies:\n")  
j=0  
for i in range(len):#function for printing freq  
 if j == 0:  
 f.write(str([tempfreqarray[i],count\_nonzero(tempfreqarray==tempfreqarray[i])])+"\n")  
 j = count\_nonzero(tempfreqarray==tempfreqarray[i])  
 j = j - 1  
f.write("\ntable of cumulative frequencies:\n")  
j=0  
s=0  
for i in range(len):#function for printing cumfreq  
 if j == 0:  
 s = count\_nonzero(tempfreqarray==tempfreqarray[i]) + s  
 f.write(str([tempfreqarray[i],s])+"\n")  
 j = count\_nonzero(tempfreqarray==tempfreqarray[i])  
 j = j - 1  
for i in range(len):#function for finding mode  
 count = count\_nonzero(datarray==datarray[i])  
 if(count > largestCount):  
 largestCount = count  
 valuePosition = i  
 i += count  
f.write("\nmode = "+str(datarray[valuePosition]))  
devsum = 0.0#initialises as float  
for i in range(len):  
 devsum = devsum + (float(datarray[i])-(float(datarray.sum())/len))\*\*2#half of the standard deviation formula  
f.write("\nstandard deviation = "+str(sqrt(devsum/(len-1))))#second half  
f.write("\nvariation = "+str(devsum/(len-1)))#variation  
def cusquantile(cqnda,cloc):#custom function for finding quantiles  
 return cqnda[int(floor(len\*cloc))]+(cqnda[int(ceil(len\*cloc))]-cqnda[int(floor(len\*cloc))])\*((len\*cloc) % 1)  
f.write("\nquartile deviation = "+str((cusquantile(datarray,0.75)-cusquantile(datarray,0.25))/2))  
f.close()

**Рішення 4 та 1,2,3**

**10**

****

Median = 39

table of frequencies:

[1, 3]

[10, 1]

[12, 2]

[66, 2]

[75, 1]

[97, 1]

table of cumulative frequencies:

[1, 3]

[10, 4]

[12, 6]

[66, 8]

[75, 9]

[97, 10]

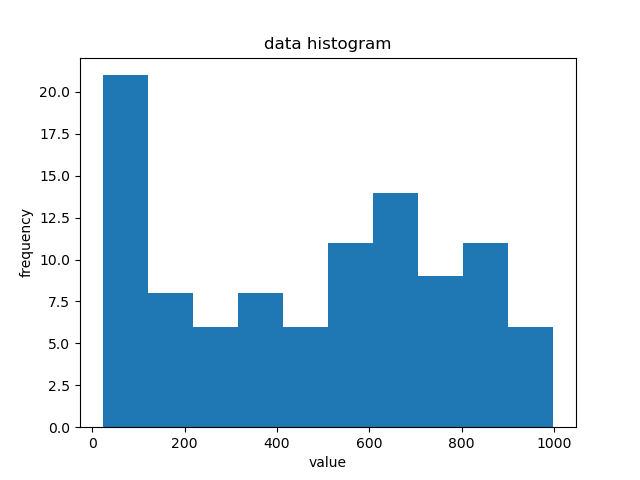
mode = 1

dispersion = 0.979591836735

standard deviation = 37.28106460091742

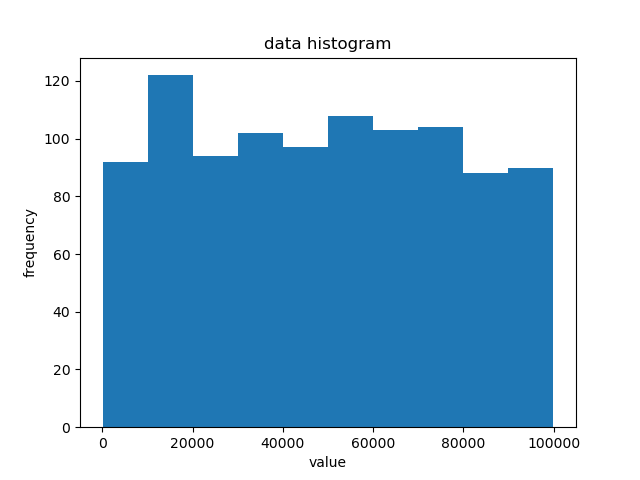
quartile deviation = 32.5

**100**



https://github.com/Andrew-Ponomarenko/laboratorni-iz-yopi/blob/main/output%20from%20input\_100.txt.txt

**1000**



https://github.com/Andrew-Ponomarenko/laboratorni-iz-yopi/blob/main/output%20from%20input\_1000.txt.txt

(файли великі тому я їх на github завантажив)

**Висновок:**

Я застосував свої знання про методи знаходження тенденцій даних за допомогою python 2.7 та бібліотек numpy та matplotlib.Після повторної перевірки результатів все виявилось правильно