|  |  |  |  |
| --- | --- | --- | --- |
| 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 № 3**  **Двовимірна статистика** | | | |
| **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 міри в двовимірній статистиці.

**Завдання:**

1. Намалюйте діаграму розсіювання для даних. Укажіть, чи існує тренд у даних.

Якщо так, то вкажіть, чи є це негативним трендом, чи позитивним.

2. Знайдіть центр ваги і коваріацію.

3. Знайти рівняння лінії регресії y від x.

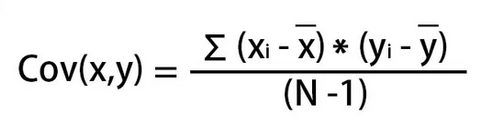
4. Розрахуйте коефіцієнт кореляції між даними.

5. Зробити висновок про залежності.

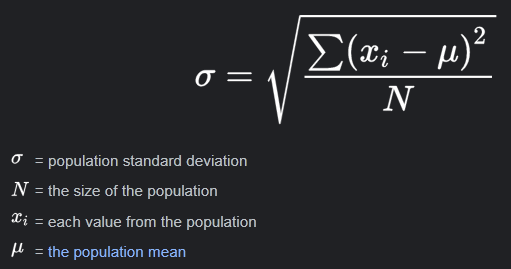
**Рішення:**

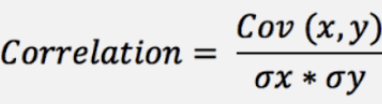
1.Використана бібліотека matplotlib.pyplot і функція scatter у яку направляються ікси та ігрики із даних. Як видно у результатах виконання у даних позитивна тенденція. Також я додав центр маси та лінію регресії з наступних завдань

2.Центр ваги CoM знаходиться за допомогою додавання всіх значень x та y окремо і потім ділення,це також є середнім значенням і я потім буду цю змінну для цього використовувати.



В змінній cov знаходиться верхня частина формули коваріації знайдена у циклі for а при виведенні цього в файл cov ділиться на кількість-1.

4.Спочатку ми знаходимо стандартне відхилення x y (sdevX;sdevY) за формулою: 

Потім з них ми з них та коваріації знаходимо корреляцію 

3.Тепер за допомогою всього що ми знайшли можна знайти рівняння лінії регресії у виді y=Ax+B.

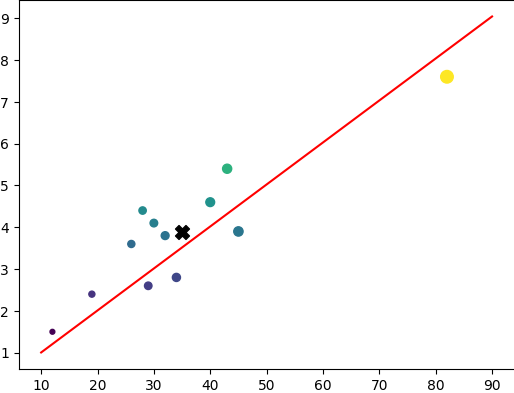
5.Як бачимо в середньому від зростання іксів зростають також і ігрики. Якби вони були незалежні нахил регресії дорівнював би нулю.

**Код програми:**

#used libraries  
from numpy import \*  
from numpy.linalg import \*  
from matplotlib.pyplot import \*  
  
#start  
filename = input() #"input\_100.txt"  
f = open(filename,"r")#opens file  
len = int(f.readline())#takes file length from first line  
datarrX = []  
datarrY = []  
templine=''  
for x in range(len):  
 templine=f.readline().split()  
 datarrX.append(float(templine[1]))  
 datarrY.append(float(templine[0].replace(',','.')))  
f.close()#closes input file  
  
#measurements to txt  
f = open("output from("+filename+").txt","w")  
#center of mass  
CoM=[0,0]  
for i in range(len):  
 CoM[0]=CoM[0]+datarrX[i]  
 CoM[1]=CoM[1]+datarrY[i]  
CoM[0]=CoM[0]/len  
CoM[1]=CoM[1]/len  
f.write("center of mass="+str(CoM)+'\n')  
#covariance  
cov=0  
cor=0  
for i in range(len):  
 cov+=(datarrX[i]-CoM[0])\*(datarrY[i]-CoM[1])  
f.write("covariance="+str(cov/(len-1))+'\n')  
devsumX=0.0  
devsumY=0.0  
for i in range(len):  
 devsumX+=(datarrX[i]-CoM[0])\*\*2  
 devsumY+=(datarrY[i]-CoM[1])\*\*2  
sdevX=sqrt(devsumX/(len-1))  
sdevY=sqrt(devsumY/(len-1))  
cor=cov/sqrt(devsumX\*devsumY)  
f.write("correlation="+str(cor)+'\n')  
#regression line Y=AX+B  
A=cor\*sdevX/sdevY  
B=CoM[0]-A\*CoM[1]  
f.write("for a regression formula of type y=a\*x+b:\nA="+str(A)+'\n')  
f.write("B="+str(B)+'\n')  
f.write("y=x\*"+str(A)+'+'+str(B)+'\n')  
  
#diagrams  
scatter(datarrX,datarrY,datarrX,datarrY)  
scatter(CoM[0],CoM[1],color=(0,0,0),s=100,marker='X')#center of mass  
  
x = linspace(10,90,300)  
y = x/A  
plot(x, y, '-r', label='y=ax+b')  
  
savefig("scatter plot from("+filename+").png",bbox\_inches='tight',pad\_inches=0)#saves file  
clf()#erases the previous plot  
  
f.close()#closes output file  
#end

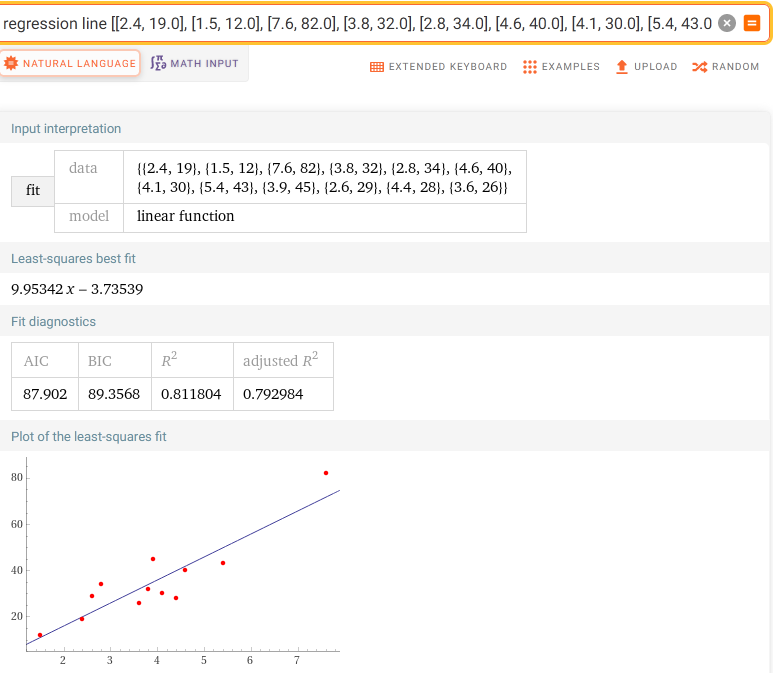
**Рішення задач 2,3,4,1**

**input\_10.txt**

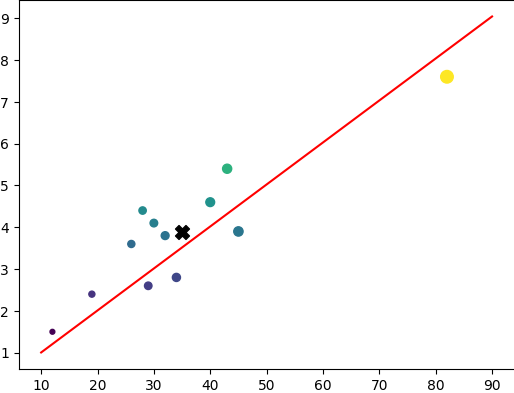


*center of mass=[35.0, 3.891666666666667]  
covariance=25.090909090909086  
correlation=0.9010014623100244  
for a regression formula of type y=a\*x+b:  
A=9.953418482344102  
B=-3.7353869271224696  
y=x\*9.953418482344102+-3.7353869271224696*

Результат перевірено на wolfram:



**input\_100.txt**

****

*center of mass=[34.5, 3.856000000000001]  
covariance=22.820202020202014  
correlation=0.9019503370715775  
for a regression formula of type y=a\*x+b:  
A=9.982043632558986  
B=-3.9907602471474632  
y=x\*9.982043632558986+-3.9907602471474632*

**Рішення задачі 5 і Висновок:**

Я застосував свої знання про методи знаходження властивостей двомірних данних з допомогою python 3.8 та бібліотек numpy,matplotlib,pandas та stemgraphic.По властивостям та графіку можна зробити висновок що другий вхідний файл є більш розширенною версією першого,з значними повтореннями,і різниця в їх властивостях є незначною і справою точності.Після повторної перевірки результатів за допомогою wolfram все є правильним.