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

Виконав:	Коваленко Владислав Олександрович	**Перевірила**:	Марцафей А. С.
Група	ІПЗ-22(2)	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Назва роботи: Двовимірна статистика

Мета: Навчитись використовувати на практиці набуті знання про міри в двовимірній статистиці.

Завдання №1:

Постановка задачі:
Намалюйте діаграму розсіювання для даних.

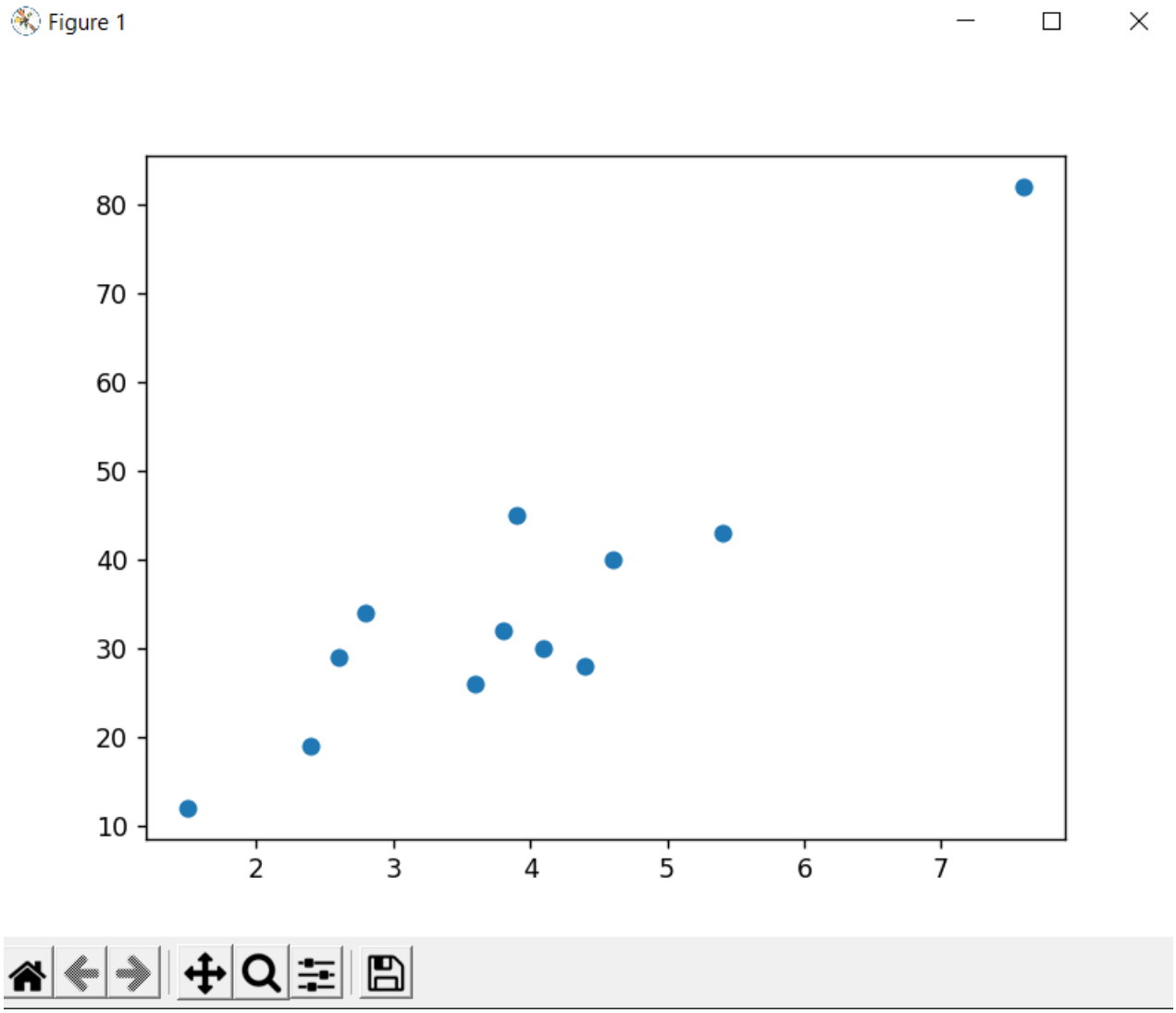
Побудова математичної моделі:
Для побудови діаграми розсіювання спочатку запишемо дані з нашого файлу в два масиви,один масив для суми покупки,а інший для проведеного часу в супермаркеті. І далі за допомогою функції scatter() бібліотеки MatPlotLib.Pyplot малюємо нашу діаграму розсіювання.І результат зберегти в файлі

Псевдокод алгориму:

```
filename = input("Введіть назву файлу: ")
outputfile = open("OutputFile.txt","w")
outputfile.writelines("Лабораторна робота №3")
with open(filename,'r') as file:
    filedata = file.read()
    filedata  = filedata.replace(',','.')
    if filename == "input_100.txt":
        str1 = filedata[4::] + "\n"
    if filename == "input_10.txt":
        str1 = filedata[3::] + "\n"
temp = ''
temp2 = ''
counter = 0;
Sum = []
Time = []
for i in range(len(str1)):
    if counter == 0:
        temp += str1[i]
    if counter != 0:
            temp2 += str1[i]
    if str1[i] == '\n':
        if(temp2 != ''):
            Time.append(temp2)
            counter = 0
            temp2 = ''
    if str1[i] == '\t':
        if(temp != ''):
            Sum.append(temp)
            counter += 1
            temp = ''
Sum = [float(x) for x in Sum]
Time = [int(x) for x in Time]

plt.scatter(Sum,Time)
plt.savefig("Task1")
plt.show()
```

Випробування алгоритму:



Завдання 2:

Постановка задачі:
Знайдіть центр ваги і коваріацію.
Побудова математичної моделі:
Для того,щоб знайти центр ваги потрібно знайти середнє арифметичне послідовності Часу та послідовності Суми покупок.А для того,щоб знайти

$$\mathrm{cov}(X,Y) = \frac{1}{n}\sum_{k=1} (x_k - M_x)(y_k - M_y)$$

$$M_x = \frac{1}{n}\sum_{k=1}^{n} x_k \ , \quad M_y = \frac{1}{n}\sum_{k=1}^{n} y_k$$

коваріацію потрібно скористатися формулою .І результат записати в файл

Псевдокод алгориму:

```
avarageSum=0
avarageTime =0;
for i in range(len(Sum)):
        avarageSum += Sum[i]
        avarageTime += Time[i]
avarageSum = avarageSum/len(Sum)
avarageTime = avarageTime/len(Sum)
outputfile.writelines("\n================Завдання 1================")
#print("G("+ str(avarageSum)+","+str(avarageTime)+")")
outputfile.writelines("\nG("+ str(avarageSum)+","+str(avarageTime)+")")

cov =0
for i in range(len(Sum)):
    cov += Sum[i]*Time[i]
cov = (1/len(Sum))*cov - avarageSum*avarageTime
#print(cov)
outputfile.writelines("\n================Завдання 2================")
outputfile.writelines("\ncov(X,Y)"+ str(cov))
```

Випробування алгоритму:

```
================Завдання 1================
G(3.891666666666667,35.0)
================Завдання 2================
cov(X,Y)22.99999999999997
```

Завдання 3:

Постановка задачі:

Знайти рівняння лініїї регресії y від x.

Побудова математичної моделі:

Для знаходження рівняння регресії y від x потрібно знати b_1 за формулою

$$Y = a + b \cdot X \quad (3.1),$$

$$b = R_{x,y}\frac{\sigma_y}{\sigma_x} = R_{x,y}\frac{S_y}{S_x}$$

$$a = M_y - b \cdot M_x \quad (3.3)$$

,деM_y та M_x це середні арифметичні,а також a.I в результаті підставити значення в наше рівняння і записати результат в файл.

Псевдокод алгоритму:

```
varX = 0;
for i in range(len(Sum)):
    varX += Sum[i]*Sum[i]
varX = (1/len(Sum))*varX - np.power(avarageSum,2)

b1 = cov/varX

b0 = avarageTime - b1*avarageSum
#print("y = "+str(b1)+"x "+str(b0))
outputfile.writelines("\n================Завдання 3================")
outputfile.writelines("\ny = "+str(b1)+"x "+str(b0))
```

Випробування алгоритму:

```
================Завдання 3================
y = 9.953418482344096x -3.7353869271224482
```

Завдання 4:

Постановка задачі:

Розрахуйте коефіцієнт кореляції між даними.

Побудова математичної моделі:

Для того,щоб розрахувати коефіцієнт кореляції між даними потрібно скористатися формулою $R_{x,y} = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y}$

$$\sigma_x^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - M_x)^2 \;,\quad \sigma_y^2 = \frac{1}{n}\sum_{k=1}^{n}(y_k - M_y)^2$$

$$M_x = \frac{1}{n}\sum_{k=1}^{n} x_k \;,\quad M_y = \frac{1}{n}\sum_{k=1}^{n} y_k \quad (2.3),$$

.І записати результат в файл

Псевдокод алгоритму:

```
R = 0
DispersionSum  = 0
DispersionTime = 0
for i in rang(Sum)):
    Dispersio += np.power(Sum[i]-avarageSum,2)
    DispersionTime +=np.power(Time[i]-avarageTime,2)
DispersionSum /= len(Sum)
DispersionTime /= len(Sum)
DispersionSum = np.sqrt(DispersionSum)
DispersionTime = np.sqrt(DispersionTime)
R = cov/(DispersionSum*DispersionTime)
#print(R)
outputfile.writelines("\n================Завдання 4================")
outputfile.writelines("\nr = "+str(R))
```

Випробування алгоритму:

```
================Завдання 4================
r = 0.9010014623100235
```

Завдання 5:

Постановка задачі:

Зробити висновок про залежності.

Побудова математичної моделі:

В залежності від коефіцієнту кореляції між даними ,запишемо в файл висновок про співвідношення послідовностей

Псевдокод алгоритму:

```
outputfile.writelines("\n================Завдання 5================")

if abs(R)==1:
    outputfile.writelines("\nДані співпадають з лінією регресії")
if abs(R) > math.sqrt(3)/2:
    outputfile.writelines("\nСтрога лінійна залежність")
if R==0:
    outputfile.writelines("\nДані незалежні")
if R<0:
    outputfile.writelines("\nЗалежніть негативна")
if R>0:
    outputfile.writelines("\nЗалежніть позитивна")
print("Файл успішно записано!")
```

Випробування алгоритму:

```
================Завдання 5================
Строга лінійна залежність
Залежніть позитивна
```

Висновок:

Виконавши цю лабораторну роботу, я навчився використовувати на практиці набуті зання про міри в двовимірній статистиці

Навчитись використовувати на практиці набуті знання про міри в двовимірній статистиці.