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:	Ковалець Олег Сергійович	**Перевірив**:	Марцафей А.С.
Група	ІПЗ-22	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

ЛАБОРАТОРНА РОБОТА № 3

ДВОВИМІРНА СТАТИСТИКА

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

Завдання

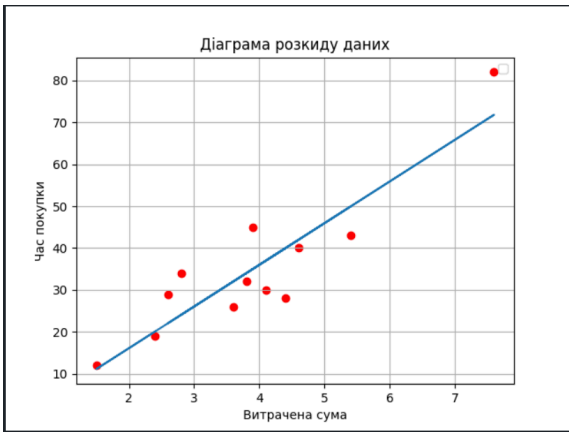
1. Намалювати діаграму розсіювання для даних.

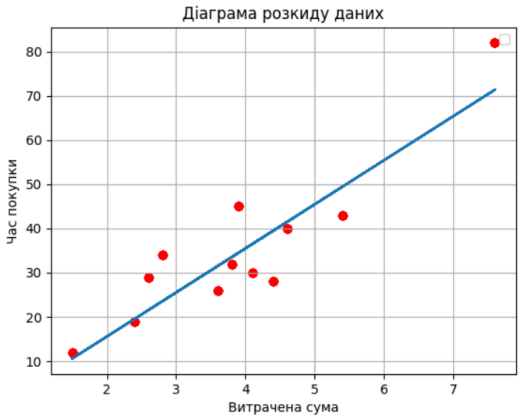
Зчитаємо поданий файл та розіб'ємо його на два масиви, за допомогою яких побудуємо діаграму розсіювання даних. Для побудови використаємо метод scatter білбіотеки matplotlib (python).

```python
import numpy as np
import matplotlib.pyplot as plt
import math


def filelist():     # Creating two arrays
    file = input("Enter a name of the file: ")
    f = open(file, "r")
    length = int(f.readline())
    sum = np.zeros(length)
    time = np.zeros(length)

    for i in range(length):
        [a, b] = f.readline().split()
        a = a.replace(',', '.')
        sum[i] = float(a)
        time[i] = int(b)

    return sum, time
```

```python
# Create plots
plt.grid(True)
plt.scatter(x, y, c = 'r')
plt.plot(x, b1*x+b0)
# Labels
plt.title("Діаграма розкиду даних")
plt.xlabel("Витрачена сума")
plt.ylabel("Час покупки")
plt.legend()
plt.show()
```

Маємо наступні діаграми для файлів **input_10.txt** та **input_100.txt** відповідно:





2. Знайдіть центр ваги і коваріацію.

Центром ваги вважається набір значень (x_, y_), де x_ та y_ - середні значення для осі x та y відповідно.

Коваріація розраховується за наступною формулою:

$$\operatorname{cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}).$$

Отже, знайдемо центр ваги та коваріацію.

```
23      # AverageX, AverageY
24      av_x, av_y = 0, 0
25      for i in range(len(x)):
26          av_x += x[i]
27          av_y += y[i]
28
29      av_x, av_y = round(av_x / len(x), 1), av_y / len(y)
30
31      covariance, var_x, var_y = 0, 0, 0
32      for i in range(len(x)):
33          covariance += (x[i] - av_x)*(y[i] - av_y)
34          var_x += pow(x[i] - av_x, 2)
35          var_y += pow(y[i] - av_y, 2)
36      # Covariance, VariationX, VariationY
37      covariance, var_x, var_y = covariance / len(x), var_x / len(x), var_y / len(y)
```

Маємо наступні значення для **input_10.txt** та **input_100.txt** відповідно:

```
Center of gravity = (3.9, 35.0)
Covariance = 22.999999999999996
```

```
Center of gravity = (3.9, 34.5)
Covariance = 22.592
```

3. Знайти рівняння лініїї регресії y від x.

Рівняння лінії регресії (y = k + mx) знаходиться за наступною формулою:

$$m \equiv b_1 = \frac{\operatorname{cov}(X,Y)}{\operatorname{Var}(X)} \text{ and } k \equiv b_0 = \bar{y} - b_1\,\bar{x}.$$

Коваріацію вже знайдено, а з нею і дисперсію для x та y в одному циклі (скріншот вище). Отже розрахуємо рівняння лінії регресії y від x.

```
39      b1 = covariance / var_x
40      b0 = av_y - b1*av_x
```

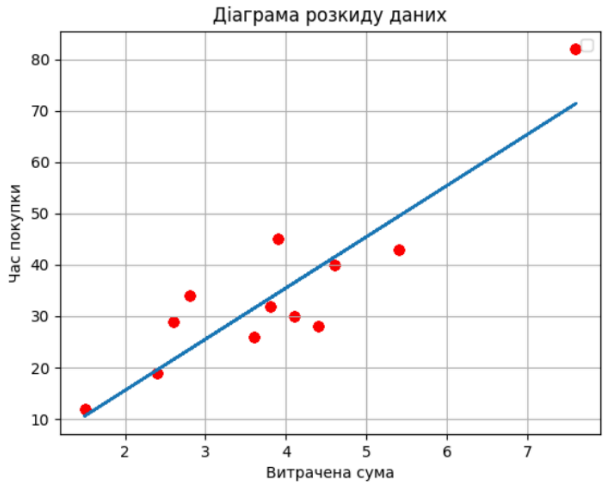
```
50      plt.plot(x, b1*x+b0)
```

Маємо наступні значення для **input_10.txt** та **input_100.txt** відповідно:

```
Regression line: y = 9.953119365308332 + (-3.817165524702496)*x
```

```
Regression line: y = 9.973512272647008 + (-4.3966978633233325)*x
```

Вставивши розраховані коефіцієнти у рівняння для matplotlib.plot, ми отримали саме ту пряму на діаграмі розсіювання даних у виводі з першого завдання.



4. Розрахуйте коефіцієнт кореляції між даними.

Знайдемо коефіцієнт кореляції за наступною формулою:

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i-\bar{x}}{s_x}\right)\left(\frac{y_i-\bar{y}}{s_y}\right),$$

де s(x) та s(y) - стандартне відхилення для x та y відповідно.

```
deviationX, deviationY, correlation = np.sqrt(var_x), np.sqrt(var_y), 0
for i in range(len(x)):
    correlation += ((x[i] - av_x) / deviationX) * ((y[i] - av_y) / deviationY)

correlation /= (len(x) - 1)
```

Маємо наступні значення для **input_10.txt** та **input_100.txt** відповідно:

```
Correlation coefficient = 0.9106715347800048
```

```
Correlation coefficient = 0.9828959169823492
```

Бачимо, що коефіцієнт кореляції знаходиться в межах від -1 до 1, як і має бути.

5. Зробити висновок про залежності.

Можемо розрахувати висновки за допомогою коду нижче:

```
writer.write("\tConclusions\n")

if correlation == 0:
    writer.write("x and y are not linearly related\n")
elif math.fabs(correlation) > np.sqrt(3) / 2:
    writer.write("x and y have strong linear relation\n")
else:
    writer.write("x and y have weak linear relation\n")

if correlation > 0:
    writer.write("The relation between x and y is positive")
elif correlation < 0:
    writer.write("The relation between x and y is negative")
```

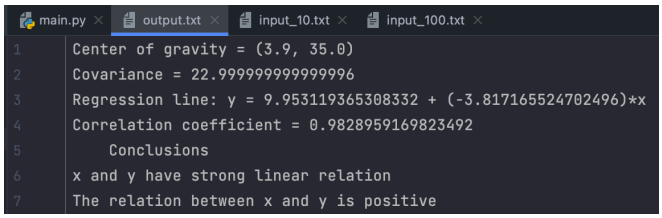
Маємо однакові висновки для двох файлів:

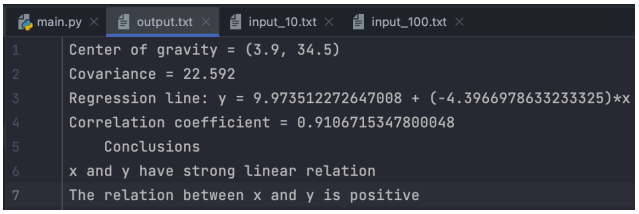
```
    Conclusions
x and y have strong linear relation
The relation between x and y is positive
```

Підсумувавши, маємо наступний код для запису результатів у файл output.txt:

```
57  # Write results in file
58  writer = open("output.txt", "w")
59  writer.write(f"Center of gravity = ({av_x}, {av_y})\n")
60  writer.write(f"Covariance = {covariance}\n")
61  writer.write(f"Regression line: y = {b1} + ({b0})*x\n")
62  writer.write(f"Correlation coefficient = {correlation}\n")
63  writer.write("\tConclusions\n")
```

Та наступний вивід:





Висновок:

Протягом даної лабораторної роботи було ознайомлено з двовимірною статистикою. Було побудовано діаграму розсіювання даних, лінію регресії та розраховано центр ваги для неї за вказаними даними. Також на основі цих даних було знайдено коваріантність та коефіцієнт кореляції. Після виконання основних було підведено підсумки. Результати записані у вихідний файл і всі вимоги виконання були дотримані.