

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:	Дармороз Аліна Олегівна	Перевірила:	
Група	ІПЗ-21(1)	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

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

**Постановка задачі.**

1. Перш за все необхідно створити функцію зчитування даних з файлу. Дані будуть перезаписані у масив, який далі буде використовуватись для обчислення статистичних даних.
2. *Намалюйте діаграму розсіювання для даних. Укажіть, чи існує тренд у даних. Якщо так, то вкажіть, чи є це негативним трендом, чи позитивним.* Для побудови діаграми розсіювання використовуємо функцію `scatter()` з бібліотеки `matplotlib` де вхідними даними буде двовимірний масив. Для обчислення тренду використовуємо формулу представлену в побудові математичної моделі, та залежно від значення (додатне, від'ємне, 0) визначаємо тренд (позитивний, негативний, немає тренду).
3. *Знайдіть центр ваги і коваріацію. Знайти рівняння лінії регресії у від x.* Для обчислення центру ваги та коваріації нам необхідний двовимірний масив на вході, обчислюючи за формулою отримуємо значення. Для знаходження рівняння лінії регресії знаходимо коваріацію та дисперсію для першого стовпчика значень, відношення цих значень буде першим коефіцієнтом.
4. *Розрахуйте коефіцієнт кореляції між даними.* На вході нам потрібен двовимірний масив за формулою обчислюємо значення коефіцієнту кореляції (Формула представлена в побудові математичної моделі)
5. На виході отримуємо результати наших обчислень виведені на консоль та записані у файл.

**Побудова математичної моделі**

1. Формула для знаходження тренду:

$$Y_t = \sum a_r x_{r+t}$$

де  $a_r$  — набір коефіцієнтів (ваг). Якщо сума коефіцієнтів дорівнює 1, то такий фільтр називають ковзаючим середнім;

2. Формула для знаходження центра ваг:

$$Y_k = (y_{ij}), i = [1; m], j = [1, n]$$

де  $y_{ij}$  — значення  $i$ -го показника в  $j$ -му досліджуваному періоді (або для  $j$ -ого досліджуваного об'єкта);

$m$  — кількість показників, що входять у  $k$ -ту групу;

$n$  — кількість досліджуваних періодів (або об'єктів).

3. Формула для знаходження коваріації:

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

, де  $E[X]$  є математичним сподіванням  $X$ , відомим також як середнє значення  $X$ ;

4. Коефіцієнт кореляції:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

## Псевдокод алгоритму.

### Зчитування даних з файлу

```
arr = []
arr.append([])
arr.append([])
file = open(name)
i = int(0)
m = int(-1)
with open(name, "r") as file:
    for line in file.readlines():
        if (i > 0):
            arr[0].append(float(line.split('
')[0].split(',')[0]+'.'+line.split('
')[0].split(',')[1]))
            arr[1].append(int(line.split('
')[1]))
        else:
            m = int(line)
            i = i + 1
file.close()
```

### Діаграма розсіювання

```
scatter(arr[0], arr[1])
```

### Визначення тренду

```
b = float(0)
sum = float(0)
avX = float(0)
avY = float(0)
for i=0 to len(arr):
    avX = avX + arr[0][i]
    avY = avY + arr[1][i]
avX = avX/m
avY = avY/m
for i=0 to len(arr):
    sum = sum + (arr[0][i]-avX)*(arr[1][i]-avY)
sum2 = float(0)
for i=0 to len(arr):
    sum2 = sum2 + (arr[0][i] - avX)*(arr[0][i] -
avX)
b = (sum)/sum2
a1 = avY - b*avX
if (b > 0):
    output("Тренд позитивний\n")
elif (b < 0):
    output("Тренд негативний\n")
else:
    output("Немає тренду\n")
x = arange(0, 10)
y = a1 + b * x
plot(x, y)
```

### Центр ваги та коваріація

```
avX = float(0)
avY = float(0)
for i=0 to len(arr):
    avX = avX + arr[0][i]
    avY = avY + arr[1][i]
avX = avX/m
avY = avY/m
output("Завдання 2\n")
output(str(round(avX, 3)) + ', ' + str(avY) + ')\n')
output("Коваріація: ")
cov = float(0)
for i in range(0, m):
    cov = cov + ((arr[0][i]-avX)*(arr[1][i]-avY))
cov = round(cov/m, 3)
output(str(cov) + '\n')
output('\n')
varx = float(0)
for i in range(0, m):
    varx = varx + (arr[0][i]*arr[0][i]-avX*avX)
varx = varx/m
b1 = cov/varx
b = avY - b1*avX
x=arange(0, m)
y = b + b1 * x
```

### Рівняння лінії регресії

```
with open(output_name, "a") as f:
    avX = float(0)
    avY = float(0)
    for i=0 to len(arr):
        avX = avX + arr[0][i]
        avY = avY + arr[1][i]
    avX = avX/m
    avY = avY/m
    cov = float(0)
    for i=0 to len(arr):
        cov = cov + ((arr[0][i]-avX)*(arr[1][i]-avY))
    cov = round(cov/m, 3)
    b = float(0)
    sum = float(0)
    varX = float(0)
    for i in range(0, m):
        varX = varX + (arr[0][i]*arr[0][i]-avX*avX)
    varX = varX/m
    b1 = cov/varX
    for i=0 to len(arr):
        sum = sum + (arr[0][i]-avX)*(arr[1][i]-avY)
    sum2 = float(0)
    for i=0 to len(arr):
        sum2 = sum2 + (arr[0][i] - avX)*(arr[0][i] -
avX)
    b = (sum)/sum2
```

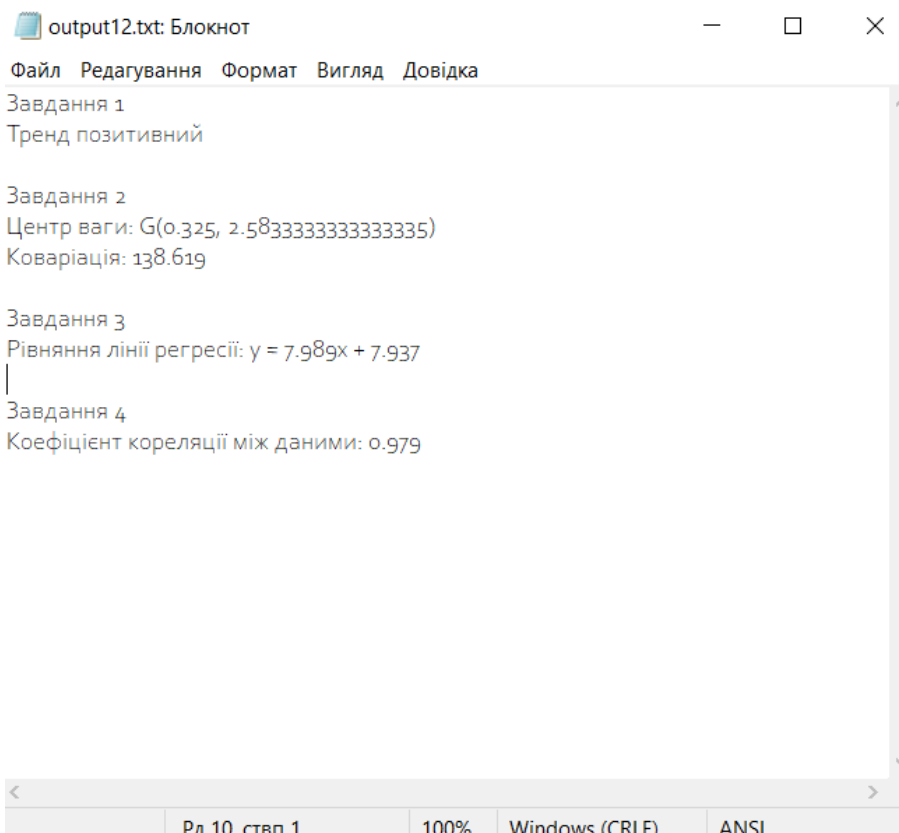
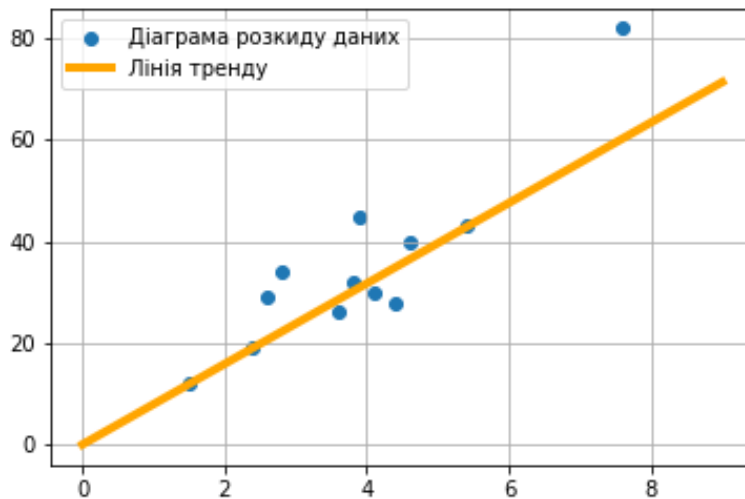
```
output(y = ' + str(round(b1, 3)) + 'x + ' +
str(round(b, 3)) + '\n')
```

### Коефіцієнт кореляції

```
output('Коефіцієнт кореляції між даними: ')
sx = float(0)
sy = float(0)
for i=0 to len(arr):
    sx = sx + (arr[0][i]-avX)*(arr[0][i]-avX)
```

```
    sx = sx/(m)
    sx = math.sqrt(sx)
    for i=0 to len(arr):
        sy = sy + (arr[1][i]-avY)*(arr[1][i]-avY)
    sy = sy/(m)
    sy = math.sqrt(sy)
    r = cov/(sx*sy)
    output(str(round(r, 3)))
```

### **Виконання алгоритму.**



**Висновок.** При виконанні цієї лабораторної роботи було розроблено алгоритм, за допомогою якого можна обчислити значення тренду, центр ваги та коваріації. Було використано на практиці знання про міри в двовимірній статистиці. Та розроблено алгоритми для їх обчислення та роботи з ними. Реалізовано обрахунок коефіцієнту

кореляції та знаходження рівняння лінії регресії. Також було побудовано діаграму розсіювання та зображено на ній лінію регресії.