

Київський національний університет ім. Т.Г.Шевченка
кафедра Програмних систем та Технологій

Звіт
до лабораторної роботи № 3
варіант №11
з дисципліни
«Ймовірнісні основи програмної інженерії»
Студента групи ІПЗ-24 групи
Липник Артем Вадимович

Лабораторна робота 3 ДВОВИМІРНА СТАТИСТИКА

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

Постановка задачі

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 у від x .
4. Розрахуйте коефіцієнт кореляції між даними.
5. Зробити висновок про залежності.

Псевдокод алгоритму

```
import math
import matplotlib.pyplot as plt
import sympy as sp
import sys

sys.stdout = open("result.txt", "w")

data = []
x = []
y = []
countData = 0
avgX = 0
avgY = 0

def setData(fileName):
    global countData
    global data
    file = open(fileName, 'r')
    for line in file:
        data.append(line.split())
```

```

tmp = []

for i in data:
    tmpJ = []
    for j in i:
        tmpJ.append(float(j))
    tmp.append(tmpJ)
data = tmp

countData = data.pop(0)

def X(data):
    global x
    for i in data:
        x.append(i[0])

def Y(data):
    global y
    for i in data:
        y.append(i[1])

def trend(data):
    if max(data) == data[len(data)-1]:
        print("Trend is positive")
    elif min(data) == data[len(data)-1]:
        print("Trend is negative")
    else:
        print("Does not have any trend")

def cov(x, y):
    global avgX, avgY
    cov = 0.0

    avgX = sum(x) / len(x)
    avgY = sum(y) / len(y)

    for i in range(len(x)):
        cov += (x[i] - avgX) * (y[i] - avgY)

```

```

cov = cov / (len(x)-1)

print("Covariance: ", cov)

def centerVag():
    m = 0
    global data
    for i in range(len(data)):
        m += data[i][0] * data[i][1]

    print("center of Weigth = ", m)

def correlation(x, y):
    global avgX, avgY
    s1 = 0.0
    s2 = 0.0
    s3 = 0.0

    for i in range(len(x)):
        s1 += (x[i] - avgX) * (y[i] - avgY)
        s2 += (x[i] - avgX) * (x[i] - avgX)
        s3 += (y[i] - avgY) * (y[i] - avgY)

    s2 = s2 * s3

    corcoef = s1/math.sqrt(s2)

    print("Correlation coefficient:", corcoef)

def lineofregression(X, Y):
    global avgX, avgY
    sumx = sum(X)
    sumy = sum(Y)
    sumxy = sum(X) * sum(Y)
    sumx2 = sum(X) * sum(X)
    sumy2 = sum(Y) * sum(Y)

    byx = (len(X) * sumxy - (sumx * sumy)) / (len(X) * sumx2 - sumx2)

```

```

x, y = sp.symbols("x,y")
line = sp.Eq(y-avgY, byx*(x-avgX))
lineX = sp.solve(line, y)
liney = sp.solve(line, x)
strlineX = str(lineX)
strliney = str(liney)
strlineX = strlineX.replace("[", "")
strlineX = strlineX.replace("]", "")
strliney = strliney.replace("[", "")
strliney = strliney.replace("]", "")
print("Line of regression of y on x")
print("x = " + strliney)
print("y = " + strlineX, "\t(y on x)")

setData("./input_10.txt")

data = sorted(data)
X(data)
Y(data)
print("Sorted data: ", data)
print("\n")
trend(data)
print("\n")
centerVag()
cov(x, y)
correlation(x, y)
print("\n")
lineofregression(x, y)
plt.scatter(x, y)
plt.show()

sys.stdout.close()

```

Випробування алгоритму

Sorted data: [[1.5, 12.0], [2.4, 19.0], [2.6, 29.0], [2.8, 34.0], [3.6, 26.0], [3.8, 32.0], [3.9, 45.0], [4.1, 30.0], [4.4, 28.0], [4.6, 40.0], [5.4, 43.0], [7.6, 82.0]]

Trend is positive

center of Weigth = 1910.5

Covariance: 25.09090909090909

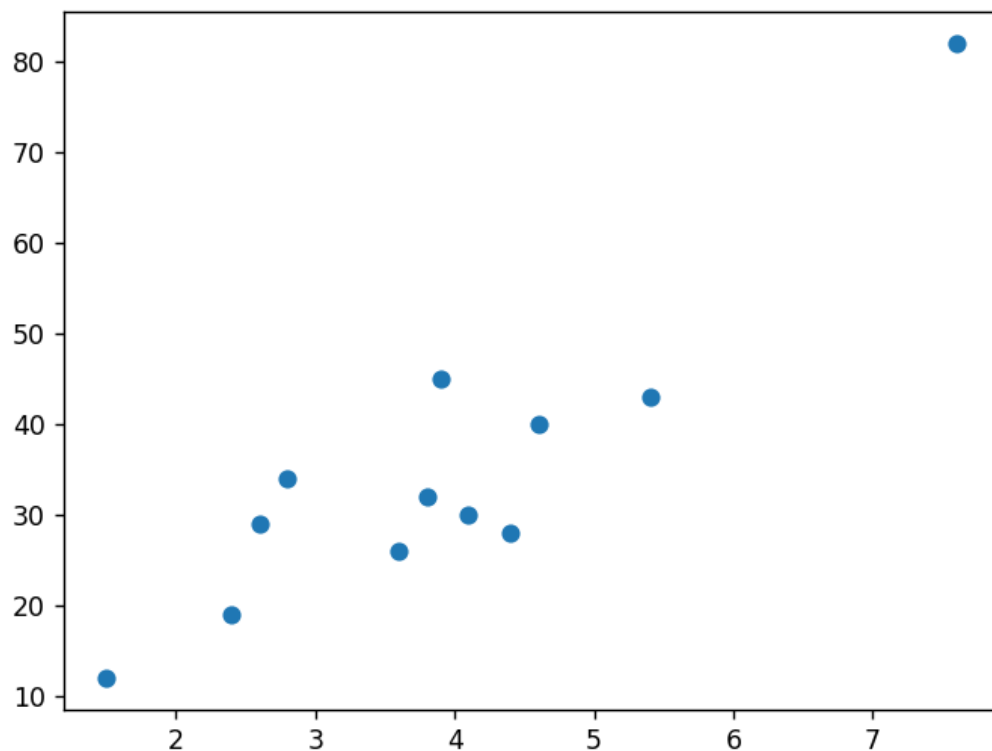
Correlation coefficient: 0.9010014623100246

Line of regression of y on x

$x = 0.111190476190476 \cdot y$

$y = 8.99357601713062 \cdot x$ (y on x)

Figure 1



Висновок: Проробивши лабораторну роботу я навчився використовувати на практиці набуті знання про міри в двовимірній