Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 2

з дисципліни «МНД» на тему
«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАЛА:
студентка II курсу ФІОТ
групи ІВ-91
Бузулук М.В.
№ заліковки - 9103

ПЕРЕВІРИВ:
ас. Регіда П. Г.

Київ – 2021

Мета: провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії..

Варіант завдання:

№ варіанту	X1		X2	
	min	max	min	max
103	-20	30	30	80

Лістинг програми:

```
import random as rd
import numpy as np
import prettytable as pt
from numpy.linalg import det


X1 = np.array([-1, -1])
X2 = np.array([1, -1])
X3 = np.array([-1, 1])

Ymax = (30-103)*10
Ymin = (20-103)*10

X1max = 30
X1min = -20
X2max = 80
X2min = 30

m = 5
while True:
    Y1 = np.array([rd.randint(Ymin, Ymax) for i in range(m)])
    Y2 = np.array([rd.randint(Ymin, Ymax) for i in range(m)])
    Y3 = np.array([rd.randint(Ymin, Ymax) for i in range(m)])

    Y1mid = sum(Y1)/len(Y1)
    Y2mid = sum(Y2)/len(Y2)
    Y3mid = sum(Y3)/len(Y3)

    disp1 = 0
    disp2 = 0
    disp3 = 0
    for i in range(len(Y1)):
        disp1 += (Y1[i]- Y1mid)**2
        disp2 += (Y2[i]- Y2mid)**2
        disp3 += (Y3[i]- Y3mid)**2

    disp1 /= len(Y1)
    disp2 /= len(Y2)
```

```
    disp3 /= len(Y3)

    Maindiv = (2*(2*m-2)/(m*(m-4)))**(1/2)
    def rahall(d1, d2):
        F = 0
        if d1 >= d2:
            F = d1/d2
        else:
            F = d2/d1

        tet = ((m-2)/m)*F

        R = abs(tet-1)/Maindiv
        return F, tet, R


    F12, Tet1, R1 = rahall(disp1, disp2)
    F13, Tet2, R2 = rahall(disp1, disp3)
    F23, Tet3, R3 = rahall(disp2, disp3)

    tab = {(5, 6, 7): 2.0, (8, 9): 2.17, (10, 11): 2.29, (12, 13): 2.39, (14, 15,
 16, 17): 2.49, (18, 19, 20): 2.62}
    Rkr = 0
    for k in tab.keys():
        if m in k:
            Rkr = tab[k]
            break

    if R1 < Rkr and R2<Rkr and R3<Rkr:
        break
    m+= 1


mx1 = (X1[0]+X2[0]+X3[0])/3
mx2 = (X1[1]+X2[1]+X3[1])/3

mmidY = (Y1mid + Y2mid + Y3mid)/3

a1 = (X1[0]**2+X2[0]**2+X3[0]**2)/3
a2 = (X1[0]*X1[1]+ X2[0]*X2[1]+X3[0]*X3[1])/3
a3 = (X1[1]**2+X2[1]**2+X3[1]**2)/3

a11 = (X1[0]*Y1mid+X2[0]*Y2mid+X3[0]*Y3mid)/3
a22 = (X1[1]*Y1mid+X2[1]*Y2mid+X3[1]*Y3mid)/3

main_det = det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])
b0 = det([[mmidY, mx1, mx2], [a11, a1, a2], [a22, a2, a3]]) / main_det
b1 = det([[1, mmidY, mx2], [mx1, a11, a2], [mx2, a22, a3]]) / main_det
b2 = det([[1, mx1, mmidY], [mx1, a1, a11], [mx2, a2, a22]]) / main_det

DX1 = abs(X1max - X1min)/2
```

```
DX2 = abs(X2max - X2min)/2
X10 = (X1max + X1min)/2
X20 = (X2max + X2min)/2

a0n = b0 - b1*X10/DX1 - b2*X20/DX2
a1n = b1/DX1
a2n = b2/DX2

st = ['Y{}'.format(i+1) for i in range(m)]
table = pt.PrettyTable()
table.field_names = ['X1', 'X2', *st]


table.add_row([*X1, *Y1])
table.add_row([*X2, *Y2])
table.add_row([*X3, *Y3])
print(table)

table2 = pt.PrettyTable()
table2.field_names = ['№', 'Середніє значення Y', 'Дисперсія Y', 'F', 'Theta', 'R
', 'Основне відхилення']
table2.add_row(['1', round(Y1mid, 3), round(disp1, 3), round(F12,3), round(Tet1,3
), round(R1,3), round(Maindiv, 3)])
table2.add_row(['2', round(Y2mid,3), round(disp2,3), round(F13,3), round(Tet2,3),
 round(R2,3), round(Maindiv, 3)])
table2.add_row(['3', round(Y3mid,3), round(disp3,3), round(F23,3), round(Tet3,3),
 round(R3,3), round(Maindiv, 3)])
print(table2)

print('\nНормалізоване рівняння:')
print('y = {} + {} * x1 + {} * x2'.format(round(b0, 3), round(b1, 3), round(b2, 3
)))
print('Перевірка:')

result_y = b0 + b1 * X1[0] + b2 * X1[1]
print('{} + {} * {} + {} * {} = {}'.format(round(b0, 3), round(b1, 3), X1[0], rou
nd(b2, 3), X1[1], round(result_y, 3)))

result_y = b0 + b1 * X2[0] + b2 * X2[1]
print('{} + {} * {} + {} * {} = {}'.format(round(b0, 3), round(b1, 3), X2[0], rou
nd(b2, 3), X2[1], round(result_y, 3)))

result_y = b0 + b1 * X3[0] + b2 * X3[1]
print('{} + {} * {} + {} * {} = {}'.format(round(b0, 3), round(b1, 3), X3[0], rou
nd(b2, 3), X3[1], round(result_y, 3)))

X1 = [X1min, X2min]
X2 = [X1max, X2min]
X3 = [X1min, X2max]

print('\nНатуралізоване рівняння:')
```

```
print('y = {} + {} * x1 + {} * x2'.format(round(a0n, 3), round(a1n, 3), round(a2n
, 3)))
print('Перевірка:')
result_y = a0n + a1n * X1[0] + a2n * X1[1]
print('{} + {} * {} + {} * {} = {}'.format(round(a0n, 3), round(a1n, 3), X1[0],
                                     round(a2n, 3), X1[1], round(result_y, 3)))

result_y = a0n + a1n * X2[0] + a2n * X2[1]
print('{} + {} * {} + {} * {} = {}'.format(round(a0n, 3), round(a1n, 3), X2[0],
                                     round(a2n, 3), X2[1], round(result_y, 3)))

result_y = a0n + a1n * X3[0] + a2n * X3[1]
print('{} + {} * {} + {} * {} = {}'.format(round(a0n, 3), round(a1n, 3), X3[0],
                                     round(a2n, 3), X3[1], round(result_y, 3)))
```

Результат роботи програми:

```
+----+----+------+------+------+------+------+
| X1 | X2 |  Y1  |  Y2  |  Y3  |  Y4  |  Y5  |
+----+----+------+------+------+------+------+
| -1 | -1 | -754 | -752 | -744 | -829 | -803 |
| 1  | -1 | -801 | -807 | -829 | -793 | -826 |
| -1 | 1  | -786 | -763 | -739 | -782 | -830 |
+----+----+------+------+------+------+------+
+---+--------------------+------------+-------+-------+-------+-------------------+
| № | Середніє значення Y | Дисперсія Y |   F   | Theta |   R   | Основне відхилення |
+---+--------------------+------------+-------+-------+-------+-------------------+
| 1 |       -776.4       |  1124.24   | 5.685 | 3.411 | 1.348 |       1.789       |
| 2 |       -811.2       |   197.76   | 1.246 | 0.748 | 0.141 |       1.789       |
| 3 |       -780.0       |   902.0    | 4.561 | 2.737 | 0.971 |       1.789       |
+---+--------------------+------------+-------+-------+-------+-------------------+

Нормалізоване рівняння:
y = -795.6 + -17.4 * x1 + -1.8 * x2
Перевірка:
-795.6 + -17.4 * -1 + -1.8 * -1 = -776.4
-795.6 + -17.4 * 1 + -1.8 * -1 = -811.2
-795.6 + -17.4 * -1 + -1.8 * 1 = -780.0

Натуралізоване рівняння:
y = -788.16 + -0.696 * x1 + -0.072 * x2
Перевірка:
-788.16 + -0.696 * -20 + -0.072 * 30 = -776.4
-788.16 + -0.696 * 30 + -0.072 * 30 = -811.2
-788.16 + -0.696 * -20 + -0.072 * 80 = -780.0
```

Контрольні питання:

1. Що таке регресійні поліноми і де вони застосовуються?
   В теорії планування експерименту найважливішою частиною є оцінка результатів вимірів. При цьому використовують апроксимуючі поліноми, за допомогою яких ми можемо описати нашу функцію. В ТПЕ ці поліноми отримали спеціальну назву - регресійні поліноми, а їх знаходження та аналіз - регресійний аналіз.
2. Визначення однорідності дисперсії.
   Однорідність дисперсії – необхідна умова підтвердження гіпотези про забезпечення нормального закону розподілу вимірюваної величини при обраній кількості повторів m та ймовірності p.
3. Що називається повним факторним експериментом?
   Повний факторний експеримент – експеримент, у якому використовуються всі можливі комбінації рівнів факторів (r^k)