

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 № 3**

з дисципліни «Методи оптимізації та планування експерименту»

на тему

**«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ  
ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»**

**ВИКОНАВ:**  
студент II курсу ФІОТ  
групи ІВ-81  
Федорусов Іван  
Варіант: 127

**ПЕРЕВІРИВ:**  
Регіда П. Г.

## Мета:

Провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

## Варіант:

№ <sub>варіанта</sub>	X <sub>1</sub>		X <sub>2</sub>		X <sub>3</sub>	
	min	max	min	max	min	max
127	-40	20	-25	10	-25	-10

$$x_{cp\max} = (x_{1\max} + x_{2\max} + x_{3\max}) / 3 = (20 + 10 - 10) / 3 = 6.67$$

$$x_{cp\min} = (x_{1\min} + x_{2\min} + x_{3\min}) / 3 = (-40 - 25 - 25) / 3 = -30$$

$$y_{\max} = 200 + x_{cp\max} = 206.67$$

$$y_{\min} = 200 + x_{cp\min} = 170$$

## Лістинг програми:

```
import random, numpy, scipy.stats, functools
```

```
while True:
```

```
    # m - кількість дослідів
```

```
    n, m = 7, 8
```

```
    x_cp = [(-40 - 25 - 25) / 3, (20 + 10 - 10) / 3]           # min, max
```

```
    y_m = [int(200 + x_cp[0]), int(200 + x_cp[1])]             # min, max
```

```
    # Матриця планування з нормованими значеннями при k = 3
```

```
    x_n = [  
        [1, -1, -1, -1],  
        [1, -1, 1, 1],  
        [1, 1, -1, 1],  
        [1, 1, 1, -1],  
        [1, -1, -1, 1],  
        [1, -1, 1, -1],  
        [1, 1, -1, -1],  
        [1, 1, 1, 1]  
    ]
```

```
    x_range = [  
        (-40, 20),  
        (-25, 10),  
        (-25, -10)  
    ]
```

```
    # Створюємо матриці з випадковими числами
```

```
    y = numpy.zeros(shape=(n, m))
```

```
    for i in range(n):
```

```
        for j in range(m):
```

```
            y[i][j] = random.randint(y_m[0], y_m[1])
```

```
    # Знаходимо середнє значення функції
```

```
    y_cp = [round(sum(i) / len(i), 2) for i in y]
```

```
    x_n = x_n[:len(y)]
```

```
    # Коли стали відомі всі дані, можемо знайти дисперсію
```

```
    disp = []
```

```

for i in range(n):
    disp.append(sum([(y_cp[i] - y[i][j]) ** 2 for j in range(m)]) / m)

x = numpy.ones(shape=(len(x_n), len(x_n[0])))
for i in range(len(x_n)):
    for j in range(1, len(x_n[i])):
        if x_n[i][j] == -1:
            x[i][j] = x_range[j - 1][0]
        else:
            x[i][j] = x_range[j - 1][1]

f1 = m - 1
f2 = n
f3 = f1 * f2
q = 0.05

# Проведення статистичних перевірок
t_student = functools.partial(scipy.stats.t.ppf, q=1 - 0.025)(df=f3)

print("Перевірка однорідності дисперсій за критерієм Кохрена")
q1 = q / f1
fishers_value = scipy.stats.f.ppf(q=1 - q1, dfn=f2, dfd=(f1 - 1) * f2)
G_kr = fishers_value / (fishers_value + f1 - 1) # G критичне
s = disp
Gp = max(s) / sum(s)

print("Gp =", Gp)
if Gp < G_kr:
    print("З ймовірністю", 1 - q, "дисперсії однорідні.")
else:
    print("Необхідно збільшити кількість дослідів")
    m = m + 1
    print("-" * 65)
    continue

print("-"*65)

# Перевіримо значущості коефіцієнтів за критерієм Стьюдента
S_kv = disp # S^2
s_kv_cp = sum(S_kv) / n # sum S^2

# Дослідимо статистичну оцінку дисперсії
s_Bs = (s_kv_cp / n / m) ** 0.5

Bs = [sum(1 * y for y in y_cp) / n]
for i in range(3): # 4 - ксть факторів
    Bs.append(sum(j[0] * j[1] for j in zip(x[:, i], y_cp)) / n)

ts = [round(abs(B) / s_Bs, 3) for B in Bs]
print("Перевірка значущості коефіцієнтів за критерієм Стьюдента")
print("Критерій Стьюдента:", ts)
res = [t for t in ts if t > t_student]

print("-"*65)

# Розрахунок коефіцієнтів рівняння регресії
mx = [sum(x[:, 1]) / n, sum(x[:, 2]) / n, sum(x[:, 3]) / n]
my = sum(y_cp) / n
a12 = sum([x[i][1] * x[i][2] for i in range(len(x))]) / n
a13 = sum([x[i][1] * x[i][3] for i in range(len(x))]) / n
a23 = sum([x[i][2] * x[i][3] for i in range(len(x))]) / n
a11 = sum([i ** 2 for i in x[:, 1]]) / n
a22 = sum([i ** 2 for i in x[:, 2]]) / n
a33 = sum([i ** 2 for i in x[:, 3]]) / n
a = [

```

```

sum([y_cp[i] * x[i][1] for i in range(len(x))]) / n,      # a1
sum([y_cp[i] * x[i][2] for i in range(len(x))]) / n,      # a2
sum([y_cp[i] * x[i][3] for i in range(len(x))]) / n      # a3
]

X = [[ 1, mx[0], mx[1], mx[2]],
      [mx[0], a11, a12, a13],
      [mx[1], a12, a22, a23],
      [mx[2], a13, a23, a33]]
Y = [my, a[0], a[1], a[2]]
B = [round(i, 2) for i in numpy.linalg.solve(X, Y)]
print("Рівняння регресії y =", B[0], "+", B[1], "* x1 +", B[2], "* x2 +", B[3], "* x3")

final_k = [B[ts.index(i)] for i in ts if i in res]
coefs = [i for i in B if i not in final_k]
print("Коефіцієнти", coefs, "статистично незначущі.\nВиключаємо їх з рівняння.")

print("-"*65)

# Виконуємо підстановку коефіцієнтів у рівняння регресії
y_new = []
for j in range(n):
    x_temp = [x[j][ts.index(k)] for k in ts if k in res]
    y_new.append(round(sum([x_temp[k] * final_k[k] for k in range(len(x_temp))])))

print("Значення \"y\" з коефіцієнтами", final_k)
print(y_new)

print("-"*65)

d = len(res)
f4 = n - d

# Перевіримо адекватність за критерієм Фішера
F_p = (m / (n - d) * sum([(y_new[i] - y_cp[i]) ** 2 for i in range(len(y))])) / (sum(dispatch) / n)

fisher = functools.partial(scipy.stats.f.ppf, q=1 - 0.05)
f_t = fisher(dfn=f4, dfd=f3)
print("Перевірка адекватності за критерієм Фішера")
print("Fp =", F_p, "\tf_t =", f_t)      # Табличне значення
if F_p < f_t:
    print("Математична модель адекватна експериментальним даним")
else:
    print("Математична модель не адекватна експериментальним даним")
break

```

## Контрольні питання:

### 1. Що називається дробовим факторним експериментом? \

Дробовий факторний експеримент – це частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови лінійної моделі.

### 2. Для чого потрібно розрахункове значення Кохрена?.

Розрахункове значення Кохрена використовують для перевірки однорідності дисперсій.

3. *Для чого перевіряється критерій Стьюдента?*

За допомогою критерію Стьюдента перевіряється значущість коефіцієнтів рівняння регресії.

4. *Чим визначається критерій Фішера і як його застосовувати?*

Критерій Фішера використовують при перевірці отриманого рівняння регресії досліджуваному об'єкту.