

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 № 5**

з дисципліни «МНД» на тему  
«Проведення трьохфакторного експерименту при використанні  
рівняння регресії з урахуванням квадратичних членів  
(центральний ортогональний композиційний план)»

ВИКОНАЛА:  
студентка II курсу ФІОТ  
групи ІВ-91  
Бузулук Марія  
Залікова - 9103

ПЕРЕВІРИВ:  
ас. Регіда П. Г.

**Мета:** Провести трьохфакторний експеримент з урахуванням квадратичних членів ,використовуючи центральний ортогональний композиційний план. Знайти рівняння регресії, яке буде адекватним для опису об'єкту.

**Варіант завдання:**

№ варіанту	X1		X2		X3	
	min	max	min	max	min	max
103	-1	4	-3	6	-1	9

**Лістинг програми:**

```
import random
from sklearn.linear_model import LinearRegression
from scipy.stats import f, t
from functools import partial
from pyDOE2 import *
from prettytable import PrettyTable

x_range = ((-1, 4), (-3, 6), (-1, 9))
y_min = 200 + int(sum([x[0] for x in x_range]) / 3)
y_max = 200 + int(sum([x[1] for x in x_range]) / 3)

def plan_matrix(n, m):
    y = np.zeros(shape=(n, m))
    for i in range(n):
        for j in range(m):
            y[i][j] = random.randint(y_min, y_max)

x_norm = ccdesign(3, center=(0, 1))
permutation = [2, 1, 0]
idx = np.empty_like(permutation)
idx[permutation] = np.arange(len(permutation))
x_norm = x_norm[:, idx]

x_norm = np.insert(x_norm, 0, 1, axis=1)

for i in range(4, 11):
    x_norm = np.insert(x_norm, i, 0, axis=1)

l = 1.215
x_norm[8][1] = -1
x_norm[9][1] = 1
x_norm[8][3] = 0
x_norm[9][3] = 0
x_norm[10][2] = -1
x_norm[11][2] = 1
x_norm[12][3] = -1
```

```

x_norm[13][3] = 1
x_norm[12][1] = 0
x_norm[13][1] = 0

def fill_tabs(x):
    for i in range(len(x)):
        x[i][4] = x[i][1] * x[i][2]
        x[i][5] = x[i][1] * x[i][3]
        x[i][6] = x[i][2] * x[i][3]
        x[i][7] = x[i][1] * x[i][3] * x[i][2]
        x[i][8] = round(x[i][1] ** 2, 3)
        x[i][9] = round(x[i][2] ** 2, 3)
        x[i][10] = round(x[i][3] ** 2, 3)
    return x

x_norm = fill_tabs(x_norm)
x = np.ones(shape=(len(x_norm), len(x_norm[0])), dtype=np.float64)
for i in range(8):
    for j in range(1, 4):
        if x_norm[i][j] == -1:
            x[i][j] = x_range[j - 1][0]
        else:
            x[i][j] = x_range[j - 1][1]

for i in range(8, len(x)):
    for j in range(1, 4):
        x[i][j] = (x_range[j - 1][0] + x_range[j - 1][1]) / 2
dx = [(x_range[i][1] - x_range[i][0]) / 2 for i in range(3)]
x[8][1] = -1 * dx[0] + x[9][1]
x[9][1] = 1 * dx[0] + x[9][1]
x[10][2] = -1 * dx[1] + x[9][2]
x[11][2] = 1 * dx[1] + x[9][2]
x[12][3] = -1 * dx[2] + x[9][3]
x[13][3] = 1 * dx[2] + x[9][3]
x = np.around(fill_tabs(x), 3)

x_table = PrettyTable()
for i in range(n):
    x_table.add_row([*x[i]])
print('Матриця планування:')
print(x_table)

x_norm_table = PrettyTable()
for i in range(n):
    x_norm_table.add_row([*x_norm[i]])
print('Нормована матриця планування:')
print(x_norm_table)

return x, y, x_norm

```

```

def coef_finding(x, y, norm=False):
    skm = LinearRegression(fit_intercept=False)
    skm.fit(x, y)
    b = skm.coef_
    if norm == 1:
        print('\nКоефіцієнти з нормованими значеннями:')
    else:
        print('\nКоефіцієнти:')
    b = [round(i, 3) for i in b]
    print(b)
    print(y)
    print('\nЗначення рівняння зі знайденими коефіцієнтами:\n{}'.format(np.dot(x,
b)))
    return b

def regression(x, b):
    y = sum([x[i] * b[i] for i in range(len(x))])
    return y

def s_kv(y, y_aver, n, m):
    res = []
    for i in range(n):
        s = sum([(y_aver[i] - y[i][j]) ** 2 for j in range(m)]) / m
        res.append(round(s, 3))
    return res

def checkFull(x, y, b, n, m):
    print('\nСтатистичні перевірки:')
    f1 = m - 1
    f2 = n
    f3 = f1 * f2
    q = 0.05

    kohren = {2:3346, 3:2758, 4:2419, 5:2159, 6:2034, 7:1911, 8:1815, 9:1736, 10:
1671}
    g_kr = kohren.get(f1)/10000

    student = partial(t.ppf, q=1 - q)
    t_student = student(df=f3)

    y_aver = [round(sum(i) / len(i), 3) for i in y]
    print('\nСередні значення Y:', y_aver)
    disp = s_kv(y, y_aver, n, m)
    print('Дисперсія Y:', disp)

    f1 = m - 1
    f2 = n
    q = 0.05

```

```

skv = s_kv(y, y_aver, n, m)
gp = max(skv) / sum(skv)

print(f'\nКритерій Кохрена:\ngp = {gp}')
if gp < g_kr:
    print('Дисперсія однорідна')
else:
    print("Дисперсія неоднорідна")
    m += 1
    start(n, m)

skv = s_kv(y, y_aver, n, m)
skv_aver = sum(skv) / n
sbs_tmp = (skv_aver / n / m) ** 0.5

def bs(x, y_aver, n):
    res = [sum(1 * y for y in y_aver) / n]
    for i in range(len(x[0])):
        b = sum(j[0] * j[1] for j in zip(x[:, i], y_aver)) / n
        res.append(b)
    return res

bs_tmp = bs(x[:, 1:], y_aver, n)
ts = [round(abs(b) / sbs_tmp, 3) for b in bs_tmp]

print('\nКритерій Стюдента:\n{:}'.format(ts))
res = [t for t in ts if t > t_student]
final_k = [b[i] for i in range(len(ts)) if ts[i] in res]
print('\nКоефіцієнти {} незначимі'.format([round(i, 3) for i in b if i not in
final_k]))
y_new = []
for j in range(n):
    y_new.append(round(regression([x[j][i] for i in range(len(ts)) if ts[i] i
n res], final_k), 3))
print('Значення функції відгуку зі значимими коефіцієнтами {}: '.format(final
_k))
print(y_new)
d = len(res)
if d >= n:
    print('\nF4 <= 0')
    return
f4 = n - d

S_ad = m / (n - d) * sum([(y_new[i] -
y_aver[i]) ** 2 for i in range(len(y))])
skv = s_kv(y, y_aver, n, m)
skv_aver = sum(skv) / n
f_p = S_ad / skv_aver

fisher = partial(f.ppf, q = 1 - q)

```

```
f_t = fisher(dfm=f4, dfd=f3)
print('\nКритерій Фішера:')
print('fp =', f_p)
print('ft =', f_t)
if f_p < f_t:
    print('Математична модель адекватна')
else:
    print('Математична модель неадекватна')

def start(n, m):
    x, y, x_norm = plan_matrix(n, m)
    y_aver = [round(sum(i) / len(i), 3) for i in y]
    b = coef_finding(x, y_aver)
    checkFull(x_norm, y, b, n, m)

start(15, 3)
```

## Результат роботи програми:

Матриця планування:											
Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8	Field 9	Field 10	Field 11	
1.0	-1.0	-3.0	-1.0	3.0	1.0	3.0	-3.0	1.0	9.0	1.0	
1.0	-1.0	-3.0	9.0	3.0	-9.0	-27.0	27.0	1.0	9.0	81.0	
1.0	-1.0	6.0	-1.0	-6.0	1.0	-6.0	6.0	1.0	36.0	1.0	
1.0	-1.0	6.0	9.0	-6.0	-9.0	54.0	-54.0	1.0	36.0	81.0	
1.0	4.0	-3.0	-1.0	-12.0	-4.0	3.0	12.0	16.0	9.0	1.0	
1.0	4.0	-3.0	9.0	-12.0	36.0	-27.0	-108.0	16.0	9.0	81.0	
1.0	4.0	6.0	-1.0	24.0	-4.0	-6.0	-24.0	16.0	36.0	1.0	
1.0	4.0	6.0	9.0	24.0	36.0	54.0	216.0	16.0	36.0	81.0	
1.0	-1.538	1.5	4.0	-2.306	-6.15	6.0	-9.225	2.364	2.25	16.0	
1.0	4.538	1.5	4.0	6.806	18.15	6.0	27.225	20.589	2.25	16.0	
1.0	1.5	-3.968	4.0	-5.951	6.0	-15.87	-23.805	2.25	15.741	16.0	
1.0	1.5	6.968	4.0	10.451	6.0	27.87	41.805	2.25	48.546	16.0	
1.0	1.5	1.5	-2.075	2.25	-3.113	-3.113	-4.669	2.25	2.25	4.306	
1.0	1.5	1.5	10.075	2.25	15.112	15.112	22.669	2.25	2.25	101.506	
1.0	1.5	1.5	4.0	2.25	6.0	6.0	9.0	2.25	2.25	16.0	
Нормована матриця планування:											
Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8	Field 9	Field 10	Field 11	
1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	-1.0	1.0	1.0	1.0	
1.0	-1.0	-1.0	1.0	1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	
1.0	-1.0	1.0	-1.0	-1.0	1.0	-1.0	1.0	1.0	1.0	1.0	
1.0	-1.0	1.0	1.0	-1.0	-1.0	1.0	-1.0	1.0	1.0	1.0	
1.0	1.0	-1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	1.0	1.0	
1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	-1.0	1.0	1.0	1.0	
1.0	1.0	1.0	-1.0	1.0	-1.0	-1.0	-1.0	1.0	1.0	1.0	
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
1.0	-1.215	0.0	0.0	-0.0	-0.0	0.0	-0.0	1.476	0.0	0.0	
1.0	1.215	0.0	0.0	0.0	0.0	0.0	0.0	1.476	0.0	0.0	
1.0	0.0	-1.215	0.0	-0.0	0.0	-0.0	-0.0	0.0	1.476	0.0	
1.0	0.0	1.215	0.0	0.0	0.0	0.0	0.0	0.0	1.476	0.0	
1.0	0.0	0.0	-1.215	0.0	-0.0	-0.0	-0.0	0.0	0.0	1.476	
1.0	0.0	0.0	1.215	0.0	0.0	0.0	0.0	0.0	0.0	1.476	
1.0	0.0	0.0	0.0	-1.215	0.0	-0.0	-0.0	0.0	0.0	1.476	
1.0	0.0	0.0	1.215	0.0	0.0	0.0	0.0	0.0	0.0	1.476	
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Коефіцієнти:											
[203.127, -0.233, -0.1, -0.046, 0.045, 0.013, -0.01, -0.007, 0.039, -0.016, 0.005]											
[203.667, 203.667, 201.333, 201.0, 203.0, 204.667, 203.0, 201.333, 204.333, 201.667, 202.0, 202.333, 203.0, 202.667, 202.667]											
Значення рівняння зі знайденими коефіцієнтами:											
[203.745 203.645 202.035 201.665 202.32 203.92											
202.95 201.13 203.108405 202.874262 203.041734 201.264674											
202.920824 202.711233 202.6315 ]											
Статистичні перевірки:											
Середні значення Y: [203.667, 203.667, 201.333, 201.0, 203.0, 204.667, 203.0, 201.333, 204.333, 201.667, 202.0, 202.333, 203.0, 202.667, 202.667]											
Дисперсія Y: [0.889,											