

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

Лабораторна робота №4  
з дисципліни МНД  
на тему:  
**Проведення трьохфакторного експерименту  
при використанні рівняння регресії з урахуванням ефекту  
взаємодії»**

Виконав:  
студент групи ІВ-93  
Королевич Б.В.  
Варіант: 310

Перевірив:  
Регіда П.Г

Київ 2021

## Варіант №310

310	-20	15	-35	10	10	20
-----	-----	----	-----	----	----	----

### Код програми

```
import random
import numpy as np
from numpy.linalg import solve
from scipy.stats import f,t

# Варіант №310
def lab4(m, n):
    x1min = -20
    x1max = 15
    x2min = -35
    x2max = 10
    x3min = 10
    x3max = 20

    # максимальне та мінімальне значення
    y_max = 200 + (x1max + x2max + x3max) / 3
    y_min = 200 + (x1min + x2min + x3min) / 3

    # матриця ПФЕ
    xn = [[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]]

    x1x2_norm, x1x3_norm, x2x3_norm, x1x2x3_norm = [0] * n, [0] * n, [0] * n, [0] * n

    for i in range(n):
        x1x2_norm[i] = xn[1][i] * xn[2][i]
        x1x3_norm[i] = xn[1][i] * xn[3][i]
        x2x3_norm[i] = xn[2][i] * xn[3][i]
        x1x2x3_norm[i] = xn[1][i] * xn[2][i] * xn[3][i]

    # заповнення у(генерація)
    y1 = [random.randint(int(y_min), int(y_max)) for i in range(n)]
    y2 = [random.randint(int(y_min), int(y_max)) for i in range(n)]
    y3 = [random.randint(int(y_min), int(y_max)) for i in range(n)]

    # матриця планування
    y_matrix = [[y1[0], y2[0], y3[0]],
                 [y1[1], y2[1], y3[1]],
                 [y1[2], y2[2], y3[2]],
                 [y1[3], y2[3], y3[3]],
                 [y1[4], y2[4], y3[4]],
                 [y1[5], y2[5], y3[5]],
                 [y1[6], y2[6], y3[6]],
                 [y1[7], y2[7], y3[7]]]

    # вивід даних за допомогою цикла
    print("Матриця планування у :")
    for i in range(n):
        print(y_matrix[i])

    x0 = [1, 1, 1, 1, 1, 1, 1, 1]

    # заміна -1 на x1_min, 1 на x1_max
    x1 = [-20, -20, 15, 15, -20, -20, 15, 15]
```

```

# заміна -1 на x2_мін, 1 на x2_макс
x2 = [-35, 10, -35, 10, -35, 10, -35, 10]

# заміна -1 на x3_мін, 1 на x3_макс
x3 = [10, 20, 20, 10, 20, 10, 10, 20]

# заповнення нулями x1x2, x1x3, x1x2x3
x1x2, x1x3, x2x3, x1x2x3 = [0] * n, [0] * n, [0] * n, [0] * n
# заповнення x1x2, x1x3, x1x2x3 добутками
for i in range(n):
    x1x2[i] = x1[i] * x2[i]
    x1x3[i] = x1[i] * x3[i]
    x2x3[i] = x2[i] * x3[i]
    x1x2x3[i] = x1[i] * x2[i] * x3[i]
# середні y
Y_average = []
for i in range(len(y_matrix)):
    Y_average.append(np.mean(y_matrix[i], axis=0))

# формуємо списки b і a
list_for_b = [xn[0], xn[1], xn[2], xn[3], x1x2_norm, x1x3_norm, x2x3_norm,
x1x2x3_norm]
list_for_a = list(zip(x0, x1, x2, x3, x1x2, x1x3, x2x3, x1x2x3))

# вивід матриці планування X
print("Матриця планування X:")
for i in range(n):
    print(list_for_a[i])
# нормовані фактори b_i
bi = []
for k in range(n):
    S = 0
    for i in range(n):
        S += (list_for_b[k][i] * Y_average[i]) / n
    bi.append(round(S, 3))
# розрахунок ai(система рівнянь) через функцію solve, вивід рівняння регресії
ai = [round(i, m) for i in solve(list_for_a, Y_average)]
print("Рівняння регресії: \n" "y = {} + {}*x1 + {}*x2 + {}*x3 + {}*x1x2 + {}*x1x3
+ {}*x2x3 "
      "+ {}*x1x2x3".format(ai[0],ai[1],ai[2],ai[3],ai[4],ai[5],ai[6],ai[7]))
# вивід даних
print("Рівняння регресії для нормованих факторів: \n" "y = {} + {}*x1 + {}*x2 +
{}*x3 + {}*x1x2 + {}*x1x3 + "
      " {}*x2x3 + {}*x1x2x3\n".format(bi[0], bi[1], bi[2], bi[3], bi[4], bi[5],
bi[6], bi[7]))

print("Перевірка за критерієм Кохрена:")
print("Середні значення відгуку за рядками:", "\n", + round(Y_average[0], 3),
round(Y_average[0], 3), round(Y_average[1], 3), round(Y_average[2], 3),
round(Y_average[3], 3),
      round(Y_average[4], 3), round(Y_average[5], 3), round(Y_average[6], 3),
round(Y_average[7],4))
# розрахунок дисперсій
dispersions = []
for i in range(len(y_matrix)):
    a = 0
    for k in y_matrix[i]:
        a += (k - np.mean(y_matrix[i], axis=0)) ** 2
    dispersions.append(a / len(y_matrix[i]))
# експериментально
Gp = max(dispersions) / sum(dispersions)

```

```

# теоретично
Gt = 0.5157
# перевірка однорідності дисперсій
if Gp < Gt:
    print("Дисперсія однорідна\n")
else:
    print("Дисперсія неоднорідна\n")
# критерій Стюдента
print("Перевірка значущості коефіцієнтів за критерієм Стюдента:")
sb = sum(dispersions) / len(dispersions)
sbs = (sb / (n * m)) ** 0.5

t_list = [abs(bi[i]) / sbs for i in range(0, n)]

d = 0
res = [0] * n
coef_1 = []
coef_2 = []
F3 = (m - 1) * n
# перевірка значущості коефіцієнтів(scipy)
for i in range(n):
    if t_list[i] < t.ppf(q=0.975, df=F3):
        coef_2.append(bi[i])
        res[i] = 0
    else:
        coef_1.append(bi[i])
        res[i] = bi[i]
        d += 1

# вивід
print("Значущі коефіцієнти регресії:\n", coef_1)
print("Незначущі коефіцієнти регресії:\n", coef_2)

# значення y з коефіцієнтами регресії
y_st = []
for i in range(n):
    y_st.append(res[0] + res[1] * xn[1][i] + res[2] * xn[2][i] + res[3] *
xn[3][i] + res[4] * x1x2_norm[i] \
                + res[5] * x1x3_norm[i] + res[6] * x2x3_norm[i] + res[7] *
x1x2x3_norm[i])
    print("Значення з отриманими коефіцієнтами:\n", y_st)

# критерій Фішера
print("\nПеревірка адекватності за критерієм Фішера:")
Sad = m * sum([(y_st[i] - Y_average[i]) ** 2 for i in range(n)]) / (n - d)
Fp = Sad / sb
F4 = n - d
# перевірка за допомогою scipy
if Fp < f.ppf(q=0.95, dfn=F4, dfd=F3):
    print("Рівняння регресії адекватне при рівні значимості 0.05")
else:
    print("Рівняння регресії неадекватне при рівні значимості 0.05")

lab4(3, 8)

```

## Результат виконання

```
C:\ProgramData\Anaconda3\envs\untitled\python.exe D:/Pycharm/Lab4/Main.py
Матриця планування y :
[205, 197, 192]
[195, 194, 190]
[208, 209, 192]
[199, 203, 193]
[200, 205, 188]
[192, 198, 198]
[214, 212, 197]
[212, 213, 213]
Матриця планування X:
(1, -20, -35, 10, 700, -200, -350, 7000)
(1, -20, 10, 20, -200, -400, 200, -4000)
(1, 15, -35, 20, -525, 300, -700, -10500)
(1, 15, 10, 10, 150, 150, 100, 1500)
(1, -20, -35, 20, 700, -400, -700, 14000)
(1, -20, 10, 10, -200, -200, 100, -2000)
(1, 15, -35, 10, -525, 150, -350, -5250)
(1, 15, 10, 20, 150, 300, 200, 3000)
Рівняння регресії:
y = 193.963 + -0.244*x1 + -0.353*x2 + 0.475*x3 + -0.018*x1x2 + 0.036*x1x3 + 0.022*x2x3 + 0.001*x1x2x3
Рівняння регресії для нормованих факторів:
y = 200.792 + 4.625*x1 + -0.792*x2 + 0.792*x3 + 0.875*x1x2 + 1.625*x1x3 + 2.042*x2x3 + 2.708*x1x2x3

Перевірка за критерієм Кохрена:
Середні значення відгуку за рядками:
 198.0 198.0 193.0 203.0 198.333 197.667 196.0 207.667 212.6667
Дисперсія однорідна
```

```
Перевірка значущості коефіцієнтів за критерієм Стьюдента:
Значущі коефіцієнти регресії:
[200.792, 4.625, 2.708]
Незначущі коефіцієнти регресії:
[-0.792, 0.792, 0.875, 1.625, 2.042]
Значення з отриманими коефіцієнтами:
[193.459, 193.459, 202.709, 202.709, 198.875, 198.875, 208.125, 208.125]

Перевірка адекватності за критерієм Фішера:
Рівняння регресії адекватне при рівні значимості 0.05

Process finished with exit code 0
```