

Міністерство освіти та науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики і 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 :
Бухтій О. В.
Перевірив:
Регіда П. Г.

Київ-2020

ЛАБОРАТОРНА РОБОТА №3.

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

Завдання

	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
107	-5	15	-15	35	15	30

$$Y_{\max} = 226$$

$$Y_{\min} = 199$$

Лістинг програми

```
from math import *
from scipy.stats import f, t
import numpy as np
from _pydecimal import Decimal
import random as rnd
import pprint
x1_min = -5
x1_max = 15
x2_min = -15
x2_max = 35
x3_min = 15
x3_max = 30
y_max = 200 + int((x1_max + x2_max + x3_max) / 3)
y_min = 200 + int((x1_min + x2_min + x3_min) / 3)
p = 0.95
q = 1 - p
m = 3
N = 4
matrix = []
flag = True
while flag:
    x_matrix = [[x1_min, x2_min, x3_min],
                [x1_min, x2_max, x3_max],
                [x1_max, x2_min, x3_max],
                [x1_max, x2_max, x3_min]]
    y_matrix = [[rnd.randrange(y_min, y_max) for j in range(m)] for i in
range(len(x_matrix))]
    y_avg = [sum(y_matrix[i]) / len(y_matrix[i]) for i in
range(len(y_matrix))]
    tmp_avg = 0
    x_avg = []
    for j in range(len(x_matrix[0])):
```

```

        tmp_avg = 0
        for i in range(len(x_matrix)):
            tmp_avg += x_matrix[i][j] / len(x_matrix)
        x_avg.append(tmp_avg)
my = sum(y_avg) / len(y_avg)
a1, a2, a3, a11, a22, a33, a12, a13, a23 = 0, 0, 0, 0, 0, 0, 0, 0, 0
for i in range(len(x_matrix)):
    a1 += x_matrix[i][0] * y_avg[i] / len(x_matrix)
    a2 += x_matrix[i][1] * y_avg[i] / len(x_matrix)
    a3 += x_matrix[i][2] * y_avg[i] / len(x_matrix)
    a11 += x_matrix[i][0]**2 / len(x_matrix)
    a22 += x_matrix[i][1]**2 / len(x_matrix)
    a33 += x_matrix[i][2]**2 / len(x_matrix)
    a12 += x_matrix[i][0] * x_matrix[i][1] / len(x_matrix)
    a13 += x_matrix[i][0] * x_matrix[i][2] / len(x_matrix)
    a23 += x_matrix[i][1] * x_matrix[i][2] / len(x_matrix)
a21 = a12
a31 = a13
a32 = a23
b0_mat = [[my, x_avg[0], x_avg[1], x_avg[2]],
            [a1, a11, a12, a13],
            [a2, a21, a22, a23],
            [a3, a31, a32, a33]]
b1_mat = [[1, my, x_avg[1], x_avg[2]],
            [x_avg[0], a1, a12, a13],
            [x_avg[1], a2, a22, a23],
            [x_avg[2], a3, a32, a33]]
b2_mat = [[1, x_avg[0], my, x_avg[2]],
            [x_avg[0], a11, a1, a13],
            [x_avg[1], a21, a2, a23],
            [x_avg[2], a31, a3, a33]]
b3_mat = [[1, x_avg[0], x_avg[1], my],
            [x_avg[0], a11, a12, a1],
            [x_avg[1], a21, a22, a2],
            [x_avg[2], a31, a32, a3]]
denom_mat = [[1, x_avg[0], x_avg[1], x_avg[2]],
              [x_avg[0], a11, a12, a13],
              [x_avg[1], a21, a22, a23],
              [x_avg[2], a31, a32, a33]]
b0 = np.linalg.det(b0_mat) / np.linalg.det(denom_mat)
b1 = np.linalg.det(b1_mat) / np.linalg.det(denom_mat)
b2 = np.linalg.det(b2_mat) / np.linalg.det(denom_mat)
b3 = np.linalg.det(b3_mat) / np.linalg.det(denom_mat)
b_list = [b0, b1, b2, b3]
f1 = m - 1
f2 = N
q = 1 - p
def cohren_value(size_of_selections, qty_of_selections, significance):
    size_of_selections += 1
    partResult1 = significance / (size_of_selections - 1)
    params = [partResult1, qty_of_selections, (size_of_selections - 1 -
1) * qty_of_selections]
    fisher = f.isf(*params)
    result = fisher / (fisher + (size_of_selections - 1 - 1))
    return Decimal(result).quantize(Decimal('.0001')).__float__()
y_disp = []
for i in range(len(x_matrix)):
    tmp_disp = 0
    for j in range(m):
        tmp_disp += ((y_matrix[i][j] - y_avg[i]) ** 2) / m
    y_disp.append(tmp_disp)
# ~ Критерий Кохрена

```

```

Gp = max(y_disp) / sum(y_disp)
Gt = cohren_value(f2, f1, q)
if Gt > Gp:
    flag = False
else:
    m += 1
# ~ Рівняння регресії

for i in range(4):
    matrix.append(x_matrix[i] + y_matrix[i])
print(" X1 X2 X3 Y1 Y2 Y3 ")
pprint.pprint(matrix)
print("y_avg--"+str(y_avg))
print("x_avg--"+str(x_avg))
print("    Рівняння регресії")
print("y = {:.2f} + {:.2f}*X1 + {:.2f}*X2 + {:.2f}*X3".format(b0, b1, b2, b3))
print("    Перевірка")
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = ".format(b0, b1,
x1_min, b2, x2_min, b3, x3_min)
+ str(round(b0 + b1 * x1_min + b2 * x2_min + b3 * x3_min,2)))
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = ".format(b0, b1,
x1_min, b2, x2_max, b3, x3_max)
+ str(round(b0 + b1 * x1_min + b2 * x2_max + b3 * x3_max,2)))
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = ".format(b0, b1,
x1_max, b2, x2_min, b3, x3_max)
+ str(round(b0 + b1 * x1_max + b2 * x2_min + b3 * x3_max,2)))
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = ".format(b0, b1,
x1_max, b2, x2_max, b3, x3_min)
+ str(round(b0 + b1 * x1_max + b2 * x2_max + b3 * x3_min,2)))
# ~ Критерій Стюдента
print("    Критерій Стюдента")
f3 = f1 * f2
S2b = sum(y_disp) / (N * N * m)
Sb = sqrt(S2b)
beta0 = ( y_avg[0] + y_avg[1] + y_avg[2] + y_avg[3]) / N
beta1 = (-y_avg[0] - y_avg[1] + y_avg[2] + y_avg[3]) / N
beta2 = (-y_avg[0] + y_avg[1] - y_avg[2] + y_avg[3]) / N
beta3 = (-y_avg[0] + y_avg[1] + y_avg[2] - y_avg[3]) / N
t0 = abs(beta0) / Sb
t1 = abs(beta1) / Sb
t2 = abs(beta2) / Sb
t3 = abs(beta3) / Sb
T_list = [t0, t1, t2, t3]
T = t.ppf((1 + (1 - q)) / 2, f3)
print("T = "+str(T)+ "\nT_list = "+str(list(map(lambda x : round(x,2),T_list))))
for i in range(len(T_list)):
    if T_list[i] < T :
        T_list[i] = 0
        b_list[i] = 0
print("    Перевірка коефіцієнтів")
y_1 = b_list[0] + b_list[1] * x1_min + b_list[2] * x2_min + b_list[3] * x3_min
y_2 = b_list[0] + b_list[1] * x1_min + b_list[2] * x2_max + b_list[3] * x3_max
y_3 = b_list[0] + b_list[1] * x1_max + b_list[2] * x2_min + b_list[3] * x3_max
y_4 = b_list[0] + b_list[1] * x1_max + b_list[2] * x2_max + b_list[3] * x3_min
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = "\
.format(b_list[0],b_list[1],x1_min,b_list[2],x2_min,b_list[3],x3_min)\
+ str(round(y_1,2))+ " = "+str(round(y_avg[0],2)))
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = "\
.format(b_list[0],b_list[1],x1_min,b_list[2],x2_max,b_list[3],x3_max)\
+ str(round(y_2,2))+ " = "+str(round(y_avg[1],2)))
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = "\
.format(b_list[0],b_list[1],x1_max,b_list[2],x2_min,b_list[3],x3_max)\

```

```

+ str(round(y_3,2))+ " = "+str(round(y_avg[2],2)))
print("{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} + {:.2f}*{:.2f} = "\
.format(b_list[0],b_list[1],x1_max,b_list[2],x2_max,b_list[3],x3_min)\
+ str(round(y_4,2))+ " = "+str(round(y_avg[3],2)))
# ~ Критерій Фішера
print("      Критерій Фішера")
b_list = list(filter(lambda i : (i != 0), b_list))
d = len(b_list)
f4 = N - d # [f3][f4]
S2ad = m * ((y_1-y_avg[0])**2 + (y_2-y_avg[1])**2 + \
            (y_3-y_avg[2])**2 + (y_4-y_avg[3])**2)/f4
Fp = S2ad / S2b
Ft=f.ppf(p, f4, f3)
print('Fp = ' + str(Fp)+"\nFt = "+str(Ft))
if Fp > Ft:
    print(" Рівняння регресії неадекватне при рівні значимості
{:.2f}".format(q))
else:
    print(" Рівняння регресії адекватне при рівні значимості {:.2f}".format(q))

```

Результат виконання

```

Terminal -
File Edit View Terminal Tabs Help
Рівняння регресії
y = 200.67+0.07*X1+-0.07*X2+0.42*X3
Перевірка
200.67+0.07*-5.00+-0.07*-15.00+0.42*15.00 = 207.67
200.67+0.07*-5.00+-0.07*35.00+0.42*30.00 = 210.67
200.67+0.07*15.00+-0.07*-15.00+0.42*30.00 = 215.33
200.67+0.07*15.00+-0.07*35.00+0.42*15.00 = 205.67
Критерій Стюдента
T = 2.3060041350333704
Перевірка коефіцієнтів
200.67 + 0.00*-5.00 + 0.00*-15.00 + 0.00*15.00 = 200.67 = 207.67
200.67 + 0.00*-5.00 + 0.00*35.00 + 0.00*30.00 = 200.67 = 210.67
200.67 + 0.00*15.00 + 0.00*-15.00 + 0.00*30.00 = 200.67 = 215.33
200.67 + 0.00*15.00 + 0.00*35.00 + 0.00*15.00 = 200.67 = 205.67
Критерій Фішера
Fp = 115.13424657534819
Ft = 4.06618055135116
Рівняння регресії неадекватне при рівні значимості 0.05

-----
(program exited with code: 0)
Press return to continue

```

Висновки

Підчас виконання лабораторної роботи було реалізовано завдання . Отримані результати збігаються , отже, експеримент було поставлено правильно.