

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 наукових досліджень»
ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

ВИКОНАВ:
Студент II курсу ФІОТ
Групи ІВ-93
Королевич Б.В.

ПЕРЕВІРИВ:
асистент
Регіда П.Г.

Київ 2021 р.

Мета:

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

Варіант завдання:

Варіант	X ₁		X ₂		X ₃	
	min	max	min	max	min	max
310	-25	-5	10	60	-5	60

Лістинг програми:

```
from prettytable import PrettyTable
from random import randint
import numpy as np
import math, os, sys
from scipy.stats import t

n = 4
m = 3
N = [i + 1 for i in range(n + 1)]
x_min = [-25, 10, -5]
x_max = [-5, 60, 60]
average_x_min = round(np.average(x_min)) #Середнє мінімальне значення
average_x_max = round(np.average(x_max)) #Середнє максимальне значення

y_min = 200 + average_x_min
y_max = 200 + average_x_max

y = [[], [], [], []]
y1 = [randint(y_min, y_max) for j in range(n)]
y2 = [randint(y_min, y_max) for j in range(n)]
y3 = [randint(y_min, y_max) for j in range(n)]
y[0] = [y1[0], y2[0], y3[0]]
y[1] = [y1[1], y2[1], y3[1]]
y[2] = [y1[2], y2[2], y3[2]]
y[3] = [y1[3], y2[3], y3[3]]

x0 = [1, 1, 1, 1]
x1 = [-1, -1, 1, 1]
x2 = [-1, 1, -1, 1]
x3 = [-1, 1, 1, -1]
x1_m = [-25, -25, -5, -5]
x2_m = [10, 60, 10, 60]
x3_m = [-5, 60, 60, -5]

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

mx1 = np.average(x1_m)
mx2 = np.average(x2_m)
mx3 = np.average(x3_m)
my = np.average(av_y)

a1 = sum([x1_m[i] * av_y[i] for i in range(n)]) / n
a2 = sum([x2_m[i] * av_y[i] for i in range(n)]) / n
a3 = sum([x3_m[i] * av_y[i] for i in range(n)]) / n
```

```

a12 = sum([x1_m[i] * x2_m[i] for i in range(n)]) / n
a13 = sum([x1_m[i] * x3_m[i] for i in range(n)]) / n
a23 = sum([x2_m[i] * x3_m[i] for i in range(n)]) / n

a11 = sum([math.pow(i, 2) for i in x1_m]) / n
a22 = sum([math.pow(i, 2) for i in x2_m]) / n
a33 = sum([math.pow(i, 2) for i in x3_m]) / n
a32, a31, a21 = a23, a13, a12

def determinant3(a11, a12, a13, a21, a22, a23, a31, a32, a33):
    determinant = a11 * a22 * a33 + a12 * a23 * a31 + a32 * a21 * a13 - a13 * a22 *
a31 - a32 * a23 * a11 - a12 * a21 * a33
    return determinant

def determinant4(a11, a12, a13, a14, a21, a22, a23, a24, a31, a32, a33, a34, a41,
a42, a43, a44):
    determinant = a11 * determinant3(a22, a23, a24, a32, a33, a34, a42, a43, a44) - \
a12 * determinant3(a21, a23, a24, a31, a33, a34, a41, a43, a44) - \
a13 * determinant3(a22, a21, a24, a32, a31, a34, a42, a41, a44) - \
a14 * determinant3(a22, a23, a21, a32, a33, a31, a42, a43, a41)
    return determinant

B0 = determinant4(1, mx1, mx2, mx3,
mx1, a11, a12, a13,
mx2, a12, a22, a23,
mx3, a13, a23, a33)

B1 = determinant4(my, mx1, mx2, mx3,
a1, a11, a12, a13,
a2, a12, a22, a23,
a3, a13, a23, a33)

B2 = determinant4(1, my, mx2, mx3,
mx1, a1, a12, a13,
mx2, a2, a22, a23,
mx3, a3, a23, a33)

B3 = determinant4(1, mx1, my, mx3,
mx1, a11, a1, a13,
mx2, a12, a2, a23,
mx3, a13, a3, a33)

B4 = determinant4(1, mx1, mx2, my,
mx1, a11, a12, a1,
mx2, a12, a22, a2,
mx3, a13, a23, a3)

b0 = B1 / B0
b1 = B2 / B0
b2 = B3 / B0
b3 = B4 / B0
b = [b0, b1, b2, b3]

yr = "y = " + str(round(b[0], 3)) + " + " + str(round(b[1], 3)) + "*x1" + " + " +
str(
round(b[2], 3)) + "*x2" + " + " + str(round(b[3], 3)) + "*x3"

```

```

y_pr1 = b[0] + b[1] * x1_m[0] + b[2] * x2_m[0] + b[3] * x3_m[0]
y_pr2 = b[0] + b[1] * x1_m[1] + b[2] * x2_m[1] + b[3] * x3_m[1]
y_pr3 = b[0] + b[1] * x1_m[2] + b[2] * x2_m[2] + b[3] * x3_m[2]
y_pr4 = b[0] + b[1] * x1_m[3] + b[2] * x2_m[3] + b[3] * x3_m[3]
y_pr = [y_pr1, y_pr2, y_pr3, y_pr4]
for i in range(3):
    if round(av_y[i], 5) == round(y_pr[i], 5):
        check1 = "Отримані значення збігаються з середніми значеннями функції відгуку за рядками"
    else:
        check1 = "Отримані значення НЕ збігаються з середніми значеннями функції відгуку за рядками"

S1 = sum([math.pow((y[0][i] - av_y[i]), 2) for i in range(m)]) / m
S2 = sum([math.pow((y[1][i] - av_y[i]), 2) for i in range(m)]) / m
S3 = sum([math.pow((y[2][i] - av_y[i]), 2) for i in range(m)]) / m
S4 = sum([math.pow((y[3][i] - av_y[i]), 2) for i in range(m)]) / m
S = [S1, S2, S3, S4]

Gp = max(S) / sum(S)

Gt = 0.7679
if Gp < Gt:
    check2 = "Дисперсія однорідна з вірогідністю 95%"
else:
    print('Помилка, повторіть експеримент заново!!!')
    os.exec1(sys.executable, sys.executable, *sys.argv)

s_beta = math.sqrt(sum(S) / (n * m * m))
s2_b = sum(S) / n

t1 = abs(sum([av_y[i] * x0[i] for i in range(n)])) / (s_beta)
t2 = abs(sum([av_y[i] * x1[i] for i in range(n)])) / (s_beta)
t3 = abs(sum([av_y[i] * x2[i] for i in range(n)])) / (s_beta)
t4 = abs(sum([av_y[i] * x3[i] for i in range(n)])) / (s_beta)
T = [t1, t2, t3, t4]
T_tabl = t.ppf(q=0.975, df=9)

k = 0
for i in range(n):
    if T[i] < T_tabl:
        b[i] = 0
        k += 1

if k != 0:
    index_list = [str(i + 1) for i, x in enumerate(b) if x == 0]
    index_list = ["b" + i for i in index_list]
    deleted_koef = ', '.join(index_list) + " - коефіцієнти рівняння регресії приймаємо незначними при рівні значимості 0.05, тобто вони виключаються з рівняння. "
else:
    deleted_koef = "Всі b значимі коефіцієнти і вони залишаються в рівнянні регресії."

ys1 = b[0] + b[1] * x1_m[0] + b[2] * x2_m[0] + b[3] * x3_m[0]
ys2 = b[0] + b[1] * x1_m[1] + b[2] * x2_m[1] + b[3] * x3_m[1]
ys3 = b[0] + b[1] * x1_m[2] + b[2] * x2_m[2] + b[3] * x3_m[2]
ys4 = b[0] + b[1] * x1_m[3] + b[2] * x2_m[3] + b[3] * x3_m[3]

```

```

y_student = [ys1, ys2, ys3, ys4]

d = n - k
f4 = n - d
F = m * sum([(av_y[i] - y_student[i]) ** 2 for i in range(n)]) / (n - d)
Fp = F / (sum(S) / n)
Fisher_table = [5.3, 4.5, 4.1, 3.8]

if (Fp < Fisher_table[f4]):
    check3 = "Рівняння регресії адекватне при рівні значимості 5%"
else:
    check3 = "Рівняння регресії неадекватне при рівні значимості 5%"

print("\nРівняння регресії: y = b0 + b1*x1 + b2*x2+ b3*x3\n")
th = ["N", "X1", "X2", "X3", "Y1", "Y2", "Y3"]
columns = len(th)
rows = len(x1)
table = PrettyTable(th)
table.title = "Натуралізована матриця планування експерименту"
for i in range(rows):
    td = [N[i], x1_m[i], x2_m[i], x3_m[i], y1[i], y2[i], y3[i]]
    td_data = td[:]
    while td_data:
        table.add_row(td_data[:columns])
        td_data = td_data[columns:]
print(table)

print("\nСередній Y:\n", round(av_y[0], 3), "\n", round(av_y[1], 3), \
      "\n", round(av_y[2], 3), "\n", round(av_y[3], 3))
print("\nОтримане рівняння регресії:", yr)
print("Практичний Y:\n", round(y_pr[0], 3), "\n", round(y_pr[1], 3), \
      "\n", round(y_pr[2], 3), "\n", round(y_pr[3], 3))
print(check1)

print("")
th = ["N", "X0", "X1", "X2", "X3", "Y1", "Y2", "Y3"]
columns = len(th)
rows = len(x1)
table = PrettyTable(th)
table.title = "Нормована матриця планування експерименту."
for i in range(rows):
    td = [N[i], x0[i], x1[i], x2[i], x3[i], y1[i], y2[i], y3[i]]
    td_data = td[:]
    while td_data:
        table.add_row(td_data[:columns])
        td_data = td_data[columns:]
print(table)

print("\nДисперсії:\n d1 =", round(S[0], 3), "\n d2 =", round(S[1], 3), \
      "\n d3 =", round(S[2], 3), "\n d4 =", round(S[3], 3))
print("Критерій Кохрена: Gr = " + str(round(Gp, 3)))
print(check2)

print("\nКритерій Стюдента:\n t1 =", round(T[0], 3), "\n t2 =", round(T[1], 3), \
      "\n t3 =", round(T[2], 3), "\n t4 =", round(T[3], 3))
print(deleted_koef)
print(" y1 =", round(y_student[0], 3), "\n y2 =", round(y_student[1], 3), \
      "\n y3 =", round(y_student[2], 3), "\n y4 =", round(y_student[3], 3))

```

```
print("\nКритерій Фішера: Fp =", round(Fp, 3))
print(check3)
```

Контрольні запитання:

1. Що називається дробовим факторним експериментом?
Дробовим факторним експериментом називається експеримент з використанням частини повного факторного експерименту.
2. Для чого потрібно розрахункове значення Кохрена?
Розрахункове значення Кохрена використовують для перевірки однорідності дисперсій.
3. Для чого перевіряється критерій Стюдента?
За допомогою критерію Стюдента 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.

Результат виконання роботи:

C:\ProgramData\Anaconda3\envs\untitled\python.exe D:/Pycharm/Lab2_AMO/Main.py

Рівняння регресії: $y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3$

Натуралізована матриця планування експерименту							
N	X1	X2	X3	Y1	Y2	Y3	
1	-25	10	-5	203	234	226	
2	-25	60	60	195	217	232	
3	-5	10	60	220	194	238	
4	-5	60	-5	230	238	195	

Середній Y:

221.0
214.67
217.33
221.0

Отримане рівняння регресії: $y = 222.544 + 0.066 \cdot x_1 + -0.027 \cdot x_2 + -0.077 \cdot x_3$

Практичний Y:

221.0
214.67
217.33
221.0

Отримані значення збігаються з середніми значеннями функції відгуку за рядками

```

+-----+
| Нормована матриця планування експерименту. |
+-----+
| N | X0 | X1 | X2 | X3 | Y1 | Y2 | Y3 |
+-----+
| 1 | 1 | -1 | -1 | -1 | 203 | 234 | 226 |
| 2 | 1 | -1 | 1 | 1 | 195 | 217 | 232 |
| 3 | 1 | 1 | -1 | 1 | 220 | 194 | 238 |
| 4 | 1 | 1 | 1 | -1 | 230 | 238 | 195 |
+-----+

Дисперсії:
d1 = 257.606
d2 = 298.879
d3 = 285.166
d4 = 374.639

Критерій Кохрена: Gr = 0.308
Дисперсія однорідна з вірогідністю 95%

Критерій Стюдента:
t1 = 150.364
t2 = 0.458
t3 = 0.458
t4 = 1.72
b2, b3, b4 - коефіцієнти рівняння регресії приймаємо незначними при рівні значимості 0.05, тобто вони виключаються з рівняння.
y1 = 222.544
y2 = 222.544
y3 = 222.544
y4 = 222.544

Критерій Фішера: Fp = 0.309
Рівняння регресії адекватне при рівні значимості 5%

```

Висновок:

В даній лабораторній роботі я провів дробовий трьохфакторний експеримент з трьома статистичними перевірками і отримав коефіцієнти рівняння регресії.