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 ІІ курсу ФІОТ
групи ІО-92
Соболь Денис

ПЕРЕВІРИВ:
Регіда П. Г.

Київ – 2021

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

Варіант завдання

218	-20	30	-35	15	-20	5

Лістинг програми

main.py

```
from random import *
import numpy  as np
from criterions import *

def nearest(arr, num):
   return arr.index(min(arr, key=lambda  x: abs(x - num)))

X_var = ((-20, 30), (-35, 15), (-20, 5))
ymax  = 200 + sum([i[1] for i in X_var]) / 3
ymin  = 200 + sum([i[0] for i in X_var]) / 3

x = (
     (1, 1, 1, 1),
     (-1, -1, 1, 1),
     (-1, 1, -1, 1),
     (-1, 1, 1, -1)
)
m = 3
N = 4
X = [[X_var[i][int((x[i+1][j]+1)/2)] for j in range(N)]  for i in range(m)]
Y = [[randrange(int(ymin),  int(ymax))  for j in range(N)] for i in range(m)]

while True:

   Yavg = [sum([Y[j][i] for j in range(m)])/m  for i in range(N)]
   mx  = [sum(i)/N for i in X]
   my  = sum(Yavg)/N
   a = [sum([X[i][j]*Yavg[j]  for j in range(N)])/N for i in range(m)]
   aa = [[sum([X[k][i]*X[j][i] for i in range(N)])/N for j in range(m)]for  k in range(m)]
   forb_denom  = ([[1] + [i for i in mx]]  + [[mx[i]] + aa[i] for i in range(m)])
   forb = [my,  a[0], a[1], a[2]]
   forb_numer  = [[forb_denom[i][0:j] + [forb[i]] + forb_denom[i][j+1:] for i in range(4)] for j in
range(4)]
   b = [np.linalg.det(forb_numer[i])/np.linalg.det(forb_denom)  for i in range(N)]
   f1 = m-1
   f2 = N
```

```
   S = [sum([(Y[j][i] - Yavg[i])**2 for j in range(m)])/m for i in range(N)]
   Gp = max(S)/sum(S)

   if Gp < G_q005[nearest(forG[0], f2)][nearest(forG[0], f1)] * 10**(-4):
      break
   else:
      m += 1
      print("m = ",m,",   N = ",N,",   f1 = ",f1,",   f2 =",f2,",   Gp = ",Gp,",   Gt = ",G_q005[f2-
2][f1-1] * 10**(-4))
      print("Додаємо кількість дослідів\n")
      for i in range(len(Y)):
         Y[i].append(randrange(int(ymin),  int(ymax)))


S_B = sum(S)/N
S_b = S_B/(N*m)
sqrt_S_b = S_b**(1/2)
beta = [sum([Yavg[j]*x[i][j] for j in range(N)])/N for i in range(N)]
t = [abs(beta[i])/sqrt_S_b for i in range(N)]
f3 = f1*f2 #8
t_tabl = k_t[nearest(fort, f3)]
tzn = [i if i > t_tabl else 0 for i in t]
bzn = [b[i] if tzn[i] != 0 else 0 for i in range(len(b))]
yzn = []
for i in range(N):
    yzn.append(bzn[0] + sum([bzn[j] * X_var[j-1][int((1 + x[j][i])/2)] for j in range(1, N)]))



d = len(tzn) - tzn.count(0)
f4 = N - d
if N == d:
   print("N = d")
   exit()
Sad = m*sum([(Yavg[i] - yzn[i])**2 for i in range(N)])/(N-d)
Fp = Sad/S_b
Ft = F[nearest(forF[0], f3)][nearest(forF[1], f4)]

print('X:',X)
print('Y:',Y)

print("\n          Критерій Корхена\n")
print('Y сер.:',[round(i,4) for i in Yavg])
print('mx:',mx)
print('my:', round(my,  4))
print("a:",[round(i, 4) for i in a])
print("aa:",aa)
print("b:", [round(i, 4) for i in b])
print("f1 = %s;  f2 = %s"%(f1, f2))
print("\n\ny  = %.2f + (%.2f) * x1 + (%.2f) * x2 + (%.2f) * x3" % tuple([round(i, 4) for i in b]))
print(round(b[0], 4)," + ",round(b[1], 4)," * ",round(X_var[0][0], 4)," + ",round(b[2], 4)," *
",round(X_var[1][0], 4)," + ",round(b[3], 4)," * ",round(X_var[2][0], 4)," = ",round(b[0] + b[1] *
```

```
X_var[0][0] + b[2] * X_var[1][0] + b[3] * X_var[2][0], 4))
print(round(b[0], 4)," + ",round(b[1], 4)," * ",round(X_var[0][1], 4)," + ",round(b[2], 4)," *
",round(X_var[1][0], 4)," + ",round(b[3], 4)," * ",round(X_var[2][1], 4)," = ",round(b[0] + b[1] *
X_var[0][1] + b[2] * X_var[1][0] + b[3] * X_var[2][1], 4))
print(round(b[0], 4)," + ",round(b[1], 4)," * ",round(X_var[0][1], 4)," + ",round(b[2], 4)," *
",round(X_var[1][1], 4)," + ",round(b[3], 4)," * ",round(X_var[2][0], 4)," = ",round(b[0] + b[1] *
X_var[0][1] + b[2] * X_var[1][1] + b[3] * X_var[2][0], 4))

print("\n          Критерій Стьюдента\n")
print("S:", [round(i, 4) for i in S])
print("Gp:", round(Gp, 4))
print("Gt:", round(G_q005[nearest(forG[0], f2)][nearest(forG[1], f1)]*10**(-4), 4))
print("S_B:", round(S_B, 4))
print("S_b = ", round(S_b, 4), " sqrt_S_b =",round(sqrt_S_b, 4))
print("beta:", [round(i, 4) for i in beta])
print("t:", [round(i, 4) for i in t])
print("f3:", f3)

print("\nТабличнне знач. t:", t_tabl)
print("tzn:", [round(i, 4) for i in tzn])
print("y = %.2f + (%.2f) * x1 + (%.2f) * x2 + (%.2f) * x3" % tuple([round(i, 4) for i in bzn]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (bzn[0], bzn[1],
X_var[0][0], bzn[2], X_var[1][0], bzn[3], X_var[2][0], bzn[0] + bzn[1] * X_var[0][0] + bzn[2] *
X_var[1][0] + bzn[3] * X_var[2][0]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (bzn[0], bzn[1], X_var[0][0], bzn[3],
X_var[2][1], bzn[0] + bzn[1] * X_var[0][0] + bzn[2] * X_var[1][1] + bzn[3] * X_var[2][1]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (bzn[0], bzn[1], X_var[0][1], bzn[3],
X_var[2][1], bzn[0] + bzn[1] * X_var[0][1] + bzn[2] * X_var[1][0] + bzn[3] * X_var[2][1]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (bzn[0], bzn[1], X_var[0][1], bzn[3],
X_var[2][0], bzn[0] + bzn[1] * X_var[0][1] + bzn[2] * X_var[1][1] + bzn[3] * X_var[2][0]))

print("\n          Критерій Фішера\n")
print("d:", d)
print("f4:", f4)
print("f3:", f3)
print("Sад:", Sad)
print("Ft:", Ft)
print("Fp:", Fp)
if Fp > Ft:
    print("\nРівняння регресії неадекватне оригіналу (Fp > Ft)")
else:
    print("\nРівняння регресії адекватне оригіналу (Fp < Ft)")
```

criterions.py

```
#G-розподіл Кохрена
```

```
forG = ((2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 20, 24, 30, 40, 60, 120),
        (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 36, 144, float("inf")))

#рівень значимості q=0.05

G_q005 = (
    (9985, 9750, 9392, 9057, 8772, 8534, 8332, 8159, 8010, 7880, 7341, 6602, 5813, 5000),
    (9669, 8709, 7977, 7457, 7071, 6771, 6530, 6333, 6167, 6025, 5466, 4748, 4031, 3333),
    (9065, 7679, 6841, 6287, 5892, 5598, 5365, 5175, 5017, 4884, 4366, 3720, 3093, 2500),
    (8412, 6838, 5981, 5440, 5063, 4783, 4564, 4387, 4241, 4118, 3645, 3066, 2513, 2000),
    (7808, 6161, 5321, 4803, 4447, 4184, 3980, 3817, 3682, 3568, 3135, 2612, 2119, 1667),
    (7271, 5612, 4800, 4307, 3974, 3726, 3535, 3384, 3259, 3154, 2756, 2278, 1833, 1429),
    (6798, 5157, 4377, 3910, 3595, 3362, 3185, 3043, 2926, 2829, 2462, 2022, 1616, 1250),
    (6385, 4775, 4027, 3584, 3286, 3067, 2901, 2768, 2659, 2568, 2226, 1820, 1446, 1111),
    (6020, 4450, 3733, 3311, 3029, 2823, 2666, 2541, 2439, 2353, 2032, 1655, 1308, 1000),
    (5410, 3924, 3264, 2880, 2624, 2439, 2299, 2187, 2098, 2020, 1737, 1403, 100, 833),
    (4709, 3346, 2758, 2419, 2159, 2034, 1911, 1815, 1736, 1671, 1429, 1144, 889, 667),
    (3894, 2705, 2205, 1921, 1735, 1602, 1501, 1422, 1357, 1303, 1108, 879, 675, 500),
    (3434, 2354, 1907, 1656, 1493, 1374, 1286, 1216, 1160, 1113, 942, 743, 567, 417),
    (2929, 1980, 1593, 1377, 1237, 1137, 1061, 1002, 958, 921, 771, 604, 457, 333),
    (2370, 30, 2370, 1576, 1259, 1082, 968, 887,827, 780, 745, 713, 595, 462, 347, 250),
    (1737, 1131, 895, 766, 682, 623, 583, 552, 520, 497, 411, 316, 234, 167),
    (998, 998, 632, 495, 419, 371, 337, 312, 292, 279, 266, 218, 165, 120, 83))

#рівень значимості q=0.01

G_q001 = (
    (9999, 950, 9794, 9586, 9373, 9172, 8988, 8823, 8764, 7539, 7949, 7067, 6062, 5000),
    (9933, 9423, 8831, 8355, 7933, 7607, 7335, 7107, 6912, 6743, 6059, 5153, 4231, 3333),
    (9676, 8643, 7814, 7212, 6761, 6410, 6129, 5897, 5702, 5536, 4884, 4057, 3251, 2500),
    (9279, 7885, 6957, 6329, 5875, 5531, 5259, 5037, 4854, 4697, 4094, 3351, 2644, 2000),
    (8828, 7218, 6258, 5635, 5195, 3866, 4608, 4401, 4229, 4048, 3529, 2858, 2229, 1667),
    (8276, 664, 5685, 5080, 4659, 4347, 4105, 3911, 3751, 3616, 3105, 2494, 1929, 1429),
    (7945, 6162, 5209, 4627, 4226, 3932, 3704, 3522, 3373, 3248, 2779, 2214, 1700, 1250),
    (7544, 5727, 4810, 4251, 3870, 3592, 3378, 3207, 3067, 2950, 2514, 1992, 1521, 1111),
    (7175, 5358, 4469, 3934, 3572, 3308, 3106, 2945, 2813, 2704, 2297, 1811, 1376, 1000),
    (6528, 4751, 3919, 3428, 3099, 2861, 2680, 2535, 2419, 2320, 1961, 1535, 1157, 833),
    (5747, 4069, 3317, 2882, 2593, 2386, 2228, 2104, 2002, 1918, 1612, 1251, 934, 667),
    (47993297, 2654, 2288, 2048, 1877, 1748, 1646, 1567, 1501, 1248, 960, 709, 500),
    (4247, 2871, 2295, 1970, 1759, 1608, 1495, 1406, 1338, 1283, 1060, 810, 595, 417),
    (3632, 2412, 1913, 1635, 1454, 1327, 1232, 1157, 1100, 1054, 867, 658, 480, 333),
    (2940, 1951, 1508, 1281, 1135, 1033, 957, 898, 853, 816, 668, 503, 363, 250),
    (2151, 1371, 1069, 902, 796, 722, 668, 625, 594, 567, 461, 344, 245, 167),
    (1252, 759, 585, 489, 429, 387, 357, 334, 316, 302, 242, 178, 125, 83))


#Перевірка за критерієм Стьюдента

fort = (tuple([i for i in range(1,31)] + [float('inf')]))
k_t = [12.71, 4.303, 3.182, 2.776, 2.571, 2.447, 2.365, 2.306, 2.262, 2.228, 2.201, 2.179, 2.16,
2.145, 2.131, 2.12, 2.11, 2.101, 2.093, 2.086, 2.08, 2.074, 2.069, 2.064, 2.06, 2.056, 2.052,
2.048, 2.045, 2.042, 1.96]
```

```
#Перевірка за критерієм Фішера

forF = (tuple([i for i in range(1, 21)] + [22, 24, 26, 28, 30, 40, 60, 120, float('inf')]), (1, 2, 3, 4, 5, 6, 12, 24, float('inf')))
F = (
    (164.4, 199.5, 215.7, 224.6, 230.2, 234, 244.9, 249, 254.3),
    (18.5, 19.2, 19.2, 19.3, 19.3, 19.3, 19.4, 19.4, 19.5),
    (10.1, 9.6, 9.3, 9.1, 9, 8.9, 8.7, 8.6, 8.5),
    (7.7, 6.9, 6.6, 6.4, 6.3, 6.2, 5.9, 5.8, 5.6),
    (6.6, 5.8, 5.4, 5.2, 5.1, 5, 4.7, 4.5, 4.4),
    (6, 5.1, 4.8, 4.5, 4.4, 4.3, 4, 3.8, 3.7),
    (5.5, 4.7, 4.4, 4.1, 4, 3.9, 3.6, 3.4, 3.2),
    (5.3, 4.5, 4.1, 3.8, 3.7, 3.6, 3.3, 3.1, 2.9),
    (5.1, 4.3, 3.9, 3.6, 3.5, 3.4, 3.1, 2.9, 2.7),
    (5, 4.1, 3.7, 3.5, 3.3, 3.2, 2.9, 2.7, 2.5),
    (4.8, 4, 3.6, 3.4, 3.2, 3.1, 2.8, 2.6, 2.4),
    (4.8, 3.9, 3.5, 3.3, 3.1, 3, 2.7, 2.5, 2.3),
    (4.7, 3.8, 3.4, 3.2, 3, 2.9, 2.6, 2.4, 2.2),
    (4.6, 3.7, 3.3, 3.1, 3, 2.9, 2.5, 2.3, 2.1),
    (4.5, 3.7, 3.3, 3.1, 2.9, 2.8, 2.5, 2.3, 2.1),
    (4.5, 3.6, 3.2, 3, 2.9, 2.7, 2.4, 2.2, 2),
    (4.5, 3.6, 3.2, 3, 2.8, 2.7, 2.4, 2.2, 2),
    (4.4, 3.6, 3.2, 2.9, 2.8, 2.7, 2.3, 2.1, 1.9),
    (4.4, 3.5, 3.1, 2.9, 2.7, 2.6, 2.3, 2.1, 1.9),
    (4.4, 3.5, 3.1, 2.9, 2.7, 2.6, 2.3, 2.1, 1.9),
    (4.3, 3.4, 3, 2.8, 2.6, 2.5, 2.2, 2, 1.7),
    (4.2, 3.4, 3, 2.7, 2.6, 2.5, 2.2, 2, 1.7),
    (4.2, 3.3, 3, 2.7, 2.6, 2.4, 2.1, 1.9, 1.7),
    (4.1, 3.2, 2.9, 2.6, 2.5, 2.3, 2, 1.8, 1.5),
    (4, 3.2, 2.8, 2.5, 2.4, 2.3, 1.9, 1.7, 1.4),
    (3.9, 3.1, 2.7, 2.5, 2.3, 2.2, 1.8, 1.6, 1.3),
    (3.8, 3, 2.6, 2.4, 2.2, 2.1, 1.8, 1.5, 1)
   )
```

Результат виконання програми

```
"E:\Semestr 4\MOPE\Ла63\venv\Scripts\python.exe" "E:/Semestr 4/MOPE/Ла63/main.py"
X: [[-20, -20, 30, 30], [-35, 15, -35, 15], [-20, 5, 5, -20]]
Y: [[208, 180, 182, 179], [194, 212, 201, 203], [176, 210, 188, 181]]

          Критерій Корхена

Y сер.: [192.6667, 200.6667, 190.3333, 187.6667]
mx: [5.0, -10.0, -7.5]
my: 192.8333
a: [868.3333, -1895.0, -1412.9167]
aa: [[650.0, -50.0, -37.5], [-50.0, 725.0, 75.0], [-37.5, 75.0, 212.5]]
b: [195.7333, -0.1533, 0.0533, 0.2133]
f1 = 2;  f2 = 4


y = 195.73 + (-0.15) * x1 + (0.05) * x2 + (0.21) * x3
195.7333  +  -0.1533  *  -20  +  0.0533  *  -35  +  0.2133  *  -20  =  192.6667
195.7333  +  -0.1533  *  30  +  0.0533  *  -35  +  0.2133  *  5  =  190.3333
195.7333  +  -0.1533  *  30  +  0.0533  *  15  +  0.2133  *  -20  =  187.6667

          Критерій Стьюдента

S: [171.5556, 214.2222, 62.8889, 118.2222]
Gp: 0.3779
Gt: 0.7679
S_B: 141.7222
S_b =  11.8102  sqrt_S_b = 3.4366
beta: [192.8333, -3.8333, 1.3333, 2.6667]
t: [56.1117, 1.1154, 0.388, 0.776]
f3: 8
Табличнне знач. t: 2.306
tzn: [60.0999, 0, 0, 0]
y = 195.22 + (0.00) * x1 + (0.00) * x2 + (0.00) * x3
195.22 + (0.00) * (-20) + (0.00) * (-35) + (0.00) * (-20) = 195.2
195.22 + (0.00) * (-20) + (0.00) * (5) = 195.2
195.22 + (0.00) * (30) + (0.00) * (5) = 195.2
195.22 + (0.00) * (30) + (0.00) * (-20) = 195.2

           Критерій Фішера

d: 1
f4: 3
f3: 8
Sад: 230.2211111111172
Ft: 4.1
Fp: 21.51785374296898

Рівняння регресії неадекватне оригіналу (Fp > Ft)
```

Висновок

У ході лабораторної роботи було
- досліджено трьохфакторний експеримент з лінійним рівнянням регресії,
- використано критерій Кохрена для перевірки дисперсій на однорідність,
- використано критерій Стьюдента для перевірки нуль-гіпотези
- використано критерій Фішера перевірки адекватності гіпотези.

Результати, що отримані при перевірці, є оптимальними, отже робота виконана правильно.

Контрольні запитання

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* показує, яку частку в загальній сумі дисперсій у рядках має максимальна з них.

3. *Для чого перевіряється критерій Стьюдента?*

Критерій *Стьюдента* використовується для перевірки значущості коефіцієнтів.

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

Критерій Фішера використовується для перевірки адекватності рівняння регресії.