**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний Технічний Університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №4**

# *з дисципліни «*Методи оптимізації та планування експерименту*»*

*на тему:* «Проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії»

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-82

*Роман О.Т.*

**Перевірив:**

*Регіда П.Г.*

**Київ – 2020**





**Код програми:**

import numpy as np  
import scipy.stats  
  
  
X1\_MIN, X1\_MAX = -10, 50  
X2\_MIN, X2\_MAX = 25, 65  
X3\_MIN, X3\_MAX = -10, 15  
m = 3  
N = 8  
  
mx\_max = (X1\_MAX + X2\_MAX + X3\_MAX) / 3  
mx\_min = (X1\_MIN + X2\_MIN + X3\_MIN) / 3  
Y\_MAX = mx\_max + 200  
Y\_MIN = mx\_min + 200  
  
y\_list = np.random.randint(Y\_MIN, Y\_MAX, (N, m))  
  
x\_matrix = [  
 [X1\_MIN, X2\_MIN, X3\_MIN],  
 [X1\_MIN, X2\_MIN, X3\_MAX],  
 [X1\_MIN, X2\_MAX, X3\_MIN],  
 [X1\_MIN, X2\_MAX, X3\_MAX],  
 [X1\_MAX, X2\_MIN, X3\_MIN],  
 [X1\_MAX, X2\_MIN, X3\_MAX],  
 [X1\_MAX, X2\_MAX, X3\_MIN],  
 [X1\_MAX, X2\_MAX, X3\_MAX]  
]  
  
while True: # цикл на проверку однородности дисперсии  
 my\_list = []  
 mx1 = 0  
 mx2 = 0  
 mx3 = 0  
  
 for obj in y\_list: # создание списка my  
 my\_list.append(sum(obj) / len(obj))  
  
 for obj in x\_matrix:  
 mx1 += obj[0]  
 mx2 += obj[1]  
 mx3 += obj[2]  
  
 mx1 /= 8  
 mx2 /= 8  
 mx3 /= 8  
 my = sum(my\_list) / 8  
  
 x123i = 0; x1i = 0; x2i = 0; x3i = 0; x12i = 0; x13i = 0; x23i = 0  
 m00 = 0; m01 = 0; m02 = 0; m03 = 0; m04 = 0; m05 = 0; m06 = 0; m07 = 0; m10 = 0; m11 = 0; m12 = 0; m13 = 0; m14 = 0  
 m15 = 0; m16 = 0; m17 = 0; m20 = 0; m21 = 0; m22 = 0; m23 = 0; m24 = 0; m25 = 0; m26 = 0; m27 = 0;  
 m30 = 0; m31 = 0; m32 = 0; m33 = 0; m34 = 0; m35 = 0; m36 = 0; m37 = 0; m40 = 0; m41 = 0; m42 = 0; m43 = 0; m44 = 0; m45 = 0; m46 = 0  
 m47 = 0; m50 = 0; m51 = 0; m52 = 0; m53 = 0; m54 = 0; m55 = 0; m56 = 0; m57 = 0; m60 = 0; m61 = 0; m62 = 0; m63 = 0  
 m64 = 0; m65 = 0; m66 = 0; m67 = 0; m70 = 0; m71 = 0  
 m72 = 0; m73 = 0; m74 = 0; m75 = 0; m76 = 0; m77 = 0  
 k0 = 0; k1 = 0; k2 = 0; k3 = 0; k4 = 0; k5 = 0; k6 = 0; k7 = 0  
  
 """Coefficients"""  
 for i in range(0, 8):  
  
 x123i = x\_matrix[i][0] \* x\_matrix[i][1] \* x\_matrix[i][2]  
 x1i = x\_matrix[i][0]  
 x2i = x\_matrix[i][1]  
 x3i = x\_matrix[i][2]  
 x12i = x1i \* x2i  
 x13i = x1i \* x3i  
 x23i = x2i \* x3i  
 m00 = N  
 m01 += x1i  
 m02 += x2i  
 m03 += x3i  
 m04 += x12i  
 m05 += x13i  
 m06 += x23i  
 m07 += x123i  
  
 m10 += x1i  
 m11 += x1i ^ 2  
 m12 += x12i  
 m13 += x13i  
 m14 += x1i ^ 2 \* x2i  
 m15 += x1i ^ 2 \* x3i  
 m16 += x123i  
 m17 += x1i ^ 2 \* x23i  
  
 m20 += x2i  
 m21 += x12i  
 m22 += x2i ^ 2  
 m23 += x23i  
 m24 += x2i ^ 2 \* x1i  
 m25 += x123i  
 m26 += x2i ^ 2 \* x3i  
 m27 += x2i ^ 2 \* x13i  
  
 m30 += x3i  
 m31 += x13i  
 m32 += x23i  
 m33 += x3i ^ 2  
 m34 += x123i  
 m35 += x3i ^ 2 \* x1i  
 m36 += x3i ^ 2 \* x2i  
 m37 += x3i ^ 2 \* x12i  
  
 m40 += x12i  
 m41 += x1i ^ 2 \* x2i  
 m42 += x1i \* x2i ^ 2  
 m43 += x123i  
 m44 += x1i ^ 2 \* x2i ^ 2  
 m45 += x1i ^ 2 \* x23i  
 m46 += x13i \* x2i ^ 2  
 m47 += x2i ^ 2 \* x1i ^ 2 \* x3i  
  
 m50 += x13i  
 m51 += x1i ^ 2 \* x3i  
 m52 += x123i  
 m53 += x1i \* x3i ^ 2  
 m54 += x1i ^ 2 \* x23i  
 m55 += x1i ^ 2 \* x3i ^ 2  
 m56 += x3i ^ 2 \* x12i  
 m57 += x3i ^ 2 \* x1i ^ 2 \* x2i  
  
 m60 += x23i  
 m61 += x123i  
 m62 += x3i \* x2i ^ 2  
 m63 += x2i \* x3i ^ 2  
 m64 += x13i \* x2i ^ 2  
 m65 += x12i \* x3i ^ 2  
 m66 += x2i ^ 2 \* x3i ^ 2  
 m67 += x2i ^ 2 \* x3i ^ 2 \* x1i  
  
 m70 += x123i  
 m71 += x1i ^ 2 \* x23i  
 m72 += x13i \* x2i ^ 2  
 m73 += x12i \* x3i ^ 2  
 m74 += x1i ^ 2 \* x2i ^ 2 \* x3i  
 m75 += x1i ^ 2 \* x3i ^ 2 \* x2i  
 m76 += x1i \* x2i ^ 2 \* x3i ^ 2  
 m77 += x2i ^ 2 \* x3i ^ 2 \* x1i ^ 2  
  
 k0 += my\_list[i]  
 k1 += my\_list[i] \* x1i  
 k2 += my\_list[i] \* x2i  
 k3 += my\_list[i] \* x3i  
 k4 += my\_list[i] \* x12i  
 k5 += my\_list[i] \* x13i  
 k6 += my\_list[i] \* x23i  
 k7 += my\_list[i] \* x123i  
  
 Det = np.linalg.det([  
 [m00, m10, m20, m30, m40, m50, m60, m70],  
 [m01, m11, m21, m31, m41, m51, m61, m71],  
 [m02, m12, m22, m32, m42, m52, m62, m72],  
 [m03, m13, m23, m33, m43, m53, m63, m73],  
 [m04, m14, m24, m34, m44, m54, m64, m74],  
 [m05, m15, m25, m35, m45, m55, m65, m75],  
 [m06, m16, m26, m36, m46, m56, m66, m76],  
 [m07, m17, m27, m37, m47, m57, m67, m77]  
 ])  
  
 b0 = np.linalg.det([  
 [k0, m10, m20, m30, m40, m50, m60, m70],  
 [k1, m11, m21, m31, m41, m51, m61, m71],  
 [k2, m12, m22, m32, m42, m52, m62, m72],  
 [k3, m13, m23, m33, m43, m53, m63, m73],  
 [k4, m14, m24, m34, m44, m54, m64, m74],  
 [k5, m15, m25, m35, m45, m55, m65, m75],  
 [k6, m16, m26, m36, m46, m56, m66, m76],  
 [k7, m17, m27, m37, m47, m57, m67, m77],  
 ]) / Det  
  
 b1 = np.linalg.det([  
 [m00, k0, m20, m30, m40, m50, m60, m70],  
 [m01, k1, m21, m31, m41, m51, m61, m71],  
 [m02, k2, m22, m32, m42, m52, m62, m72],  
 [m03, k3, m23, m33, m43, m53, m63, m73],  
 [m04, k4, m24, m34, m44, m54, m64, m74],  
 [m05, k5, m25, m35, m45, m55, m65, m75],  
 [m06, k6, m26, m36, m46, m56, m66, m76],  
 [m07, k7, m27, m37, m47, m57, m67, m77]]) / Det  
  
 b2 = np.linalg.det([  
 [m00, m10, k0, m30, m40, m50, m60, m70],  
 [m01, m11, k1, m31, m41, m51, m61, m71],  
 [m02, m12, k2, m32, m42, m52, m62, m72],  
 [m03, m13, k3, m33, m43, m53, m63, m73],  
 [m04, m14, k4, m34, m44, m54, m64, m74],  
 [m05, m15, k5, m35, m45, m55, m65, m75],  
 [m06, m16, k6, m36, m46, m56, m66, m76],  
 [m07, m17, k7, m37, m47, m57, m67, m77]  
 ]) / Det  
  
 b3 = np.linalg.det([  
 [m00, m10, m20, k0, m40, m50, m60, m70],  
 [m01, m11, m21, k1, m41, m51, m61, m71],  
 [m02, m12, m22, k2, m42, m52, m62, m72],  
 [m03, m13, m23, k3, m43, m53, m63, m73],  
 [m04, m14, m24, k4, m44, m54, m64, m74],  
 [m05, m15, m25, k5, m45, m55, m65, m75],  
 [m06, m16, m26, k6, m46, m56, m66, m76],  
 [m07, m17, m27, k7, m47, m57, m67, m77]  
 ]) / Det  
  
 b12 = np.linalg.det([  
 [m00, m10, m20, m30, k0, m50, m60, m70],  
 [m01, m11, m21, m31, k1, m51, m61, m71],  
 [m02, m12, m22, m32, k2, m52, m62, m72],  
 [m03, m13, m23, m33, k3, m53, m63, m73],  
 [m04, m14, m24, m34, k4, m54, m64, m74],  
 [m05, m15, m25, m35, k5, m55, m65, m75],  
 [m06, m16, m26, m36, k6, m56, m66, m76],  
 [m07, m17, m27, m37, k7, m57, m67, m77]  
 ]) / Det  
  
 b13 = np.linalg.det([  
 [m00, m10, m20, m30, m40, k0, m60, m70],  
 [m01, m11, m21, m31, m41, k1, m61, m71],  
 [m02, m12, m22, m32, m42, k2, m62, m72],  
 [m03, m13, m23, m33, m43, k3, m63, m73],  
 [m04, m14, m24, m34, m44, k4, m64, m74],  
 [m05, m15, m25, m35, m45, k5, m65, m75],  
 [m06, m16, m26, m36, m46, k6, m66, m76],  
 [m07, m17, m27, m37, m47, k7, m67, m77]  
 ]) / Det  
  
 b23 = np.linalg.det([  
 [m00, m10, m20, m30, m40, m50, k0, m70],  
 [m01, m11, m21, m31, m41, m51, k1, m71],  
 [m02, m12, m22, m32, m42, m52, k2, m72],  
 [m03, m13, m23, m33, m43, m53, k3, m73],  
 [m04, m14, m24, m34, m44, m54, k4, m74],  
 [m05, m15, m25, m35, m45, m55, k5, m75],  
 [m06, m16, m26, m36, m46, m56, k6, m76],  
 [m07, m17, m27, m37, m47, m57, k7, m77]  
 ]) / Det  
  
 b123 = np.linalg.det([  
 [m00, m10, m20, m30, m40, m50, m60, k0],  
 [m01, m11, m21, m31, m41, m51, m61, k1],  
 [m02, m12, m22, m32, m42, m52, m62, k2],  
 [m03, m13, m23, m33, m43, m53, m63, k3],  
 [m04, m14, m24, m34, m44, m54, m64, k4],  
 [m05, m15, m25, m35, m45, m55, m65, k5],  
 [m06, m16, m26, m36, m46, m56, m66, k6],  
 [m07, m17, m27, m37, m47, m57, m67, k7]  
 ]) / Det  
  
 print("b\u2080:", "%.2f" % b0, " b\u2081:", "%.2f" % b1, " b\u2082:", "%.2f" % b2, " b\u2083:", "%.2f" % b3, " b\u2081\u2082:", "%.2f" % b12,  
 " b\u2081\u2083:", "%.2f" % b13, " b\u2081\u2082\u2083:", "%.2f" % b123)  
  
 print(  
 f"Рівняння регресії: y = {b0:.2f}{b1:+.2f}\*x\u2081{b2:+.2f}\*x\u2082{b3:+.2f}\*x\u2083{b12:+.2f}\*x\u2081\u2082{b13:+.2f}\*x\u2081\u2083{b123:+.2f}\*x\u2081\u2082\u2083")  
  
 # find dispersion  
 S2 = []  
 for i in range(len(y\_list)):  
 S2.append(((y\_list[i][0] - my\_list[i]) \*\* 2 + (y\_list[i][1] - my\_list[i]) \*\* 2 + (  
 y\_list[i][2] - my\_list[i]) \*\* 2) / 3)  
  
 """KOHREN"""  
 Gp = max(S2) / sum(S2)  
  
 m = len(y\_list[0])  
 f1 = m - 1  
 f2 = N # N=8  
 q = 0.05  
  
 Gt = [None, 0.68, 0.516, 0.438, 0.391, 0.3595, 0.3365, 0.3185, 0.3043, 0.2926, 0.2829, 0.2462, 0.2022, 0.1616,  
 0.1250]  
 print("Gt:", Gt[f1])  
  
 if Gp < Gt[f1]:  
 print("Дисперсія однорідна")  
 break  
 else:  
 print("Дисперсія не однорідна")  
 m += 1  
 for obj in y\_list:  
 obj.append(np.random.randint(Y\_MIN, Y\_MAX))  
  
x\_matrix\_normal = [  
 [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],  
]  
  
"""STUDENT"""  
  
  
def getBeta(i):  
 sum = 0  
 for j in range(N):  
 sum += my\_list[j] \* x\_matrix\_normal[j][i]  
 sum /= N  
 return sum  
  
  
S2B = sum(S2) / N  
S2beta = S2B / (N \* m)  
Sbeta = np.sqrt(S2beta)  
  
beta0 = getBeta(0)  
beta1 = getBeta(1)  
beta2 = getBeta(2)  
beta3 = getBeta(3)  
  
t\_criterion = []  
t\_criterion.append(abs(beta0) / Sbeta, )  
t\_criterion.append(abs(beta1) / Sbeta)  
t\_criterion.append(abs(beta2) / Sbeta)  
t\_criterion.append(abs(beta3) / Sbeta)  
  
t0 = abs(beta0) / Sbeta  
t1 = abs(beta1) / Sbeta  
t2 = abs(beta2) / Sbeta  
t3 = abs(beta3) / Sbeta  
  
f3 = f1 \* f2  
  
t\_tab = scipy.stats.t.ppf((1 + (1 - q)) / 2, f3)  
print("t табличне:", t\_tab)  
if t0 < t\_tab:  
 b0 = 0  
 print("t\u2080:", t0, " t0<t\_tab; b0=0")  
if t1 < t\_tab:  
 b1 = 0  
 print("t\u2081:", t1, " t\u2081<t\_tab; b\u2081=0")  
if t2 < t\_tab:  
 b2 = 0  
 print("t\u2082:", t2, " t\u2082<t\_tab; b\u2082=0")  
if t3 < t\_tab:  
 b3 = 0  
 print("t\u2083:", t3, " t\u2083<t\_tab; b\u2083=0")  
  
y\_hat = []  
for i in range(N):  
 y\_hat.append(  
 b0 + b1 \* x\_matrix[i][0] + b2 \* x\_matrix[i][1] + b3 \* x\_matrix[i][2] + b12 \* x\_matrix[i][0] \* x\_matrix[i][1] +  
 b13 \* x\_matrix[i][0] \* x\_matrix[i][2] + b123 \* x\_matrix[i][0] \* x\_matrix[i][1] \* x\_matrix[i][2])  
  
 print(f"^y{chr(8321+i)} = {b0:.2f}{b1:+.2f}\*x{chr(8321+i)}\u2081{b2:+.2f}\*x{chr(8321+i)}\u2082{b3:+.2f}\*x{chr(8321+i)}\u2083{b12:+.2f}\*x{chr(8321+i)}\u2081"  
 f"\*x{chr(8321+i)}\u2082{b13:+.2f}\*x{chr(8321+i)}\u2081\*x{chr(8321+i)}\u2083{b123:+.2f}\*x{chr(8321+i)}\u2081\*x{chr(8321+i)}\u2082\*x{chr(8321+i)}\u2083 "  
 f"= {y\_hat[i]:.2f}")  
  
"""FISHER"""  
  
d = 2  
f4 = N - d  
S2\_ad = 0  
for i in range(N):  
 S2\_ad += (m / (N - d) \* ((y\_hat[i] - my\_list[i]) \*\* 2))  
  
Fp = S2\_ad / S2B  
Ft = scipy.stats.f.ppf(1 - q, f4, f3)  
print("Fp:", Fp)  
print("Ft:", Ft)  
if Fp > Ft:  
 print("Рівняння регресії не адекватно оригіналу при рівні значимості 0,05")  
else:  
 print("Рівняння регресії адекватно оригіналу при рівні значимості 0,05")

**Результати роботи програми:**

