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 2-го курсу ФІОТ

групи ІО-83

Малашкін В’ячеслав

Номер у списку: 16

Варiант: 316

Перевірив:  
Регіда П.Г.

Київ – 2020

**Код:**

import numpy as np

x1min = -10

x1max = 50

x2min = -20

x2max = 60

x3min = 50

x3max = 55

mx\_max = (x1max + x2max + x3max) / 3

mx\_min = (x1min + x2min + x3min) / 3

y\_max = mx\_max + 200

y\_min = mx\_min + 200

y\_list = np.random.randint(y\_min, y\_max, (4, 3)) # create tab 4\*3 random 'y' in [y\_min; y\_max]

x\_matrix = [

[x1min, x2min, x3min],

[x1min, x2max, x3max],

[x1max, x2min, x3max],

[x1max, x2max, x3min]

]

my\_list = []

mx1 = 0

mx2 = 0

mx3 = 0

for obj in y\_list:

my\_list.append((obj[0]+obj[1]+obj[2])/3)

for obj in x\_matrix:

mx1 += obj[0]

mx2 += obj[1]

mx3 += obj[2]

my1 = my\_list[0]

my2 = my\_list[1]

my3 = my\_list[2]

my4 = my\_list[3]

mx1 /= 4

mx2 /= 4

mx3 /= 4

my = (my1 + my2 + my3 + my4)/4

"""Coefficients"""

a1 = (x\_matrix[0][0]\*my1 + x\_matrix[1][0]\*my2 + x\_matrix[2][0]\*my3 + x\_matrix[3][0]\*my4)/4

a2 = (x\_matrix[0][1]\*my1 + x\_matrix[1][1]\*my2 + x\_matrix[2][1]\*my3 + x\_matrix[3][1]\*my4)/4

a3 = (x\_matrix[0][2]\*my1 + x\_matrix[1][2]\*my2 + x\_matrix[2][2]\*my3 + x\_matrix[3][2]\*my4)/4

a11 = (x\_matrix[0][0]\*\*2 + x\_matrix[1][0]\*\*2 + x\_matrix[2][0]\*\*2 + x\_matrix[3][0]\*\*2)/4

a22 = (x\_matrix[0][1]\*\*2 + x\_matrix[1][1]\*\*2 + x\_matrix[2][1]\*\*2 + x\_matrix[3][1]\*\*2)/4

a33 = (x\_matrix[0][2]\*\*2 + x\_matrix[1][2]\*\*2 + x\_matrix[2][2]\*\*2 + x\_matrix[3][2]\*\*2)/4

a12 = a21 = (x\_matrix[0][0]\*x\_matrix[0][1] + x\_matrix[1][0]\*x\_matrix[1][1] +

x\_matrix[2][0]\*x\_matrix[2][1] + x\_matrix[3][0]\*x\_matrix[3][1])/4

a13 = a31 = (x\_matrix[0][0]\*x\_matrix[0][2] + x\_matrix[1][0]\*x\_matrix[1][2] +

x\_matrix[2][0]\*x\_matrix[2][2] + x\_matrix[3][0]\*x\_matrix[3][2])/4

a23 = a32 = (x\_matrix[0][1]\*x\_matrix[0][2] + x\_matrix[1][1]\*x\_matrix[1][2] +

x\_matrix[2][1]\*x\_matrix[2][2] + x\_matrix[3][1]\*x\_matrix[3][2])/4

denominator = np.linalg.det([

[1, mx1, mx2, mx3],

[mx1, a11, a12, a13],

[mx2, a12, a22, a32],

[mx3, a13, a23, a33]

])

numerator\_b0 = np.linalg.det([

[my, mx1, mx2, mx3],

[a1, a11, a12, a13],

[a2, a12, a22, a32],

[a3, a13, a23, a33]

])

numerator\_b1 = np.linalg.det([

[1, my, mx2, mx3],

[mx1, a1, a12, a13],

[mx2, a2, a22, a32],

[mx3, a3, a23, a33]

])

numerator\_b2 = np.linalg.det([

[1, mx1, my, mx3],

[mx1, a11, a1, a13],

[mx2, a12, a2, a32],

[mx3, a13, a3, a33]

])

numerator\_b3 = np.linalg.det([

[1, mx1, mx2, my],

[mx1, a11, a12, a1],

[mx2, a12, a22, a2],

[mx3, a13, a23, a3]

])

b0 = numerator\_b0/denominator

b1 = numerator\_b1/denominator

b2 = numerator\_b2/denominator

b3 = numerator\_b3/denominator

print("b0:", "%.2f" % b0, " b1:", "%.2f" % b1, " b2:", "%.2f" % b2, " b3:", "%.2f" % b3)

print(f"Рівняння регресії: y = {b0:.2f}{b1:+.2f}\*x1{b2:+.2f}\*x2{b3:+.2f}\*x3")

if (b0 + b1\*x\_matrix[0][0] + b2\*x\_matrix[0][1] + b3\*x\_matrix[0][2]) == my1:

print("b0 + b1\*X11 + b2\*X12 + b3\*X13=my1")

print("b0 + b1\*X11 + b2\*X12 + b3\*X13 =", "%.2f" % (b0 + b1\*x\_matrix[0][0] + b2\*x\_matrix[0][1] + b3\*x\_matrix[0][2]),

"| my1 =", "%.2f" % my1)

print("b0 + b1\*X21 + b2\*X22 + b3\*X23 =", "%.2f" % (b0 + b1\*x\_matrix[1][0] + b2\*x\_matrix[1][1] + b3\*x\_matrix[1][2]),

"| my2 =", "%.2f" % my2)

print("b0 + b1\*X31 + b2\*X32 + b3\*X33 =", "%.2f" % (b0 + b1\*x\_matrix[2][0] + b2\*x\_matrix[2][1] + b3\*x\_matrix[2][2]),

"| my3 =", "%.2f" % my3)

print("b0 + b1\*X41 + b2\*X42 + b3\*X43 =", "%.2f" % (b0 + b1\*x\_matrix[3][0] + b2\*x\_matrix[3][1] + b3\*x\_matrix[3][2]),

"| my4 =", "%.2f" % my4)

x\_matrix\_normal = [

[1, -1, -1, -1],

[1, -1, 1, 1],

[1, 1, -1, 1],

[1, 1, 1, -1]

]

# 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)

S2y1 = S2[0]

S2y2 = S2[1]

S2y3 = S2[2]

S2y4 = S2[3]

"""KOHREN"""

Gp = max(S2)/sum(S2)

m = len(y\_list[0])

f1 = m-1

f2 = N = len(x\_matrix)

q = 0.05

# для q = 0.05, f1 = 2, f2 = 4, Gt = 0.7679

Gt = 0.7679

if Gp < Gt:

print("Дисперсія однорідна")

else:

print("Дисперсія не однорідна")

"""STUDENT"""

S2B = sum(S2)/N

S2beta = S2B/(N\*m)

Sbeta = np.sqrt(S2beta)

beta0 = (my1\*x\_matrix\_normal[0][0] + my2\*x\_matrix\_normal[1][0] + my3\*x\_matrix\_normal[2][0] +

my4\*x\_matrix\_normal[3][0])/4

beta1 = (my1\*x\_matrix\_normal[0][1] + my2\*x\_matrix\_normal[1][1] + my3\*x\_matrix\_normal[2][1] +

my4\*x\_matrix\_normal[3][1])/4

beta2 = (my1\*x\_matrix\_normal[0][2] + my2\*x\_matrix\_normal[1][2] + my3\*x\_matrix\_normal[2][2] +

my4\*x\_matrix\_normal[3][2])/4

beta3 = (my1\*x\_matrix\_normal[0][3] + my2\*x\_matrix\_normal[1][3] + my3\*x\_matrix\_normal[2][3] +

my4\*x\_matrix\_normal[3][3])/4

t0 = abs(beta0)/Sbeta

t1 = abs(beta1)/Sbeta

t2 = abs(beta2)/Sbeta

t3 = abs(beta3)/Sbeta

f3 = f1\*f2

t\_tab = 2.306 # для значення f3 = 8, t табличне = 2,306

print(t0, t1,t2,t3)

if t0 < t\_tab:

b0 = 0

print("t0<t\_tab; b0=0")

if t1 < t\_tab:

b1 = 0

print("t1<t\_tab; b1=0")

if t2 < t\_tab:

b2 = 0

print("t2<t\_tab; b2=0")

if t3 < t\_tab:

b3 = 0

print("t3<t\_tab; b3=0")

y1\_hat = b0 + b1\*x\_matrix[0][0] + b2\*x\_matrix[0][1] + b3\*x\_matrix[0][2]

y2\_hat = b0 + b1\*x\_matrix[1][0] + b2\*x\_matrix[1][1] + b3\*x\_matrix[1][2]

y3\_hat = b0 + b1\*x\_matrix[2][0] + b2\*x\_matrix[2][1] + b3\*x\_matrix[2][2]

y4\_hat = b0 + b1\*x\_matrix[3][0] + b2\*x\_matrix[3][1] + b3\*x\_matrix[3][2]

print(f"y1\_hat = {b0:.2f}{b1:+.2f}\*x11{b2:+.2f}\*x12{b3:+.2f}\*x13 "

f"= {y1\_hat:.2f}")

print(f"y2\_hat = {b0:.2f}{b1:+.2f}\*x21{b2:+.2f}\*x22{b3:+.2f}\*x23"

f" = {y2\_hat:.2f}")

print(f"y3\_hat = {b0:.2f}{b1:+.2f}\*x31{b2:+.2f}\*x32{b3:+.2f}\*x33 "

f"= {y3\_hat:.2f}")

print(f"y4\_hat = {b0:.2f}{b1:+.2f}\*x41{b2:+.2f}\*x42{b3:+.2f}\*x43"

f" = {y4\_hat:.2f}")

"""FISHER"""

d = 2

f4 = N - d

S2\_ad = (m/(N-d))\*((y1\_hat-my1)\*\*2 + (y2\_hat-my2)\*\*2 + (y3\_hat-my3)\*\*2 + (y4\_hat-my4)\*\*2)

Fp = S2\_ad/S2B

Ft = 4.5 # для f3=8; f4=2

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")

**Вивід:**

b0: 214.72 b1: 0.04 b2: -0.33 b3: 0.50

Рівняння регресії: y = 214.72+0.04\*x1-0.33\*x2+0.50\*x3

b0 + b1\*X11 + b2\*X12 + b3\*X13 = 218.00 | my1 = 218.00

b0 + b1\*X21 + b2\*X22 + b3\*X23 = 214.00 | my2 = 214.00

b0 + b1\*X31 + b2\*X32 + b3\*X33 = 222.33 | my3 = 222.33

b0 + b1\*X41 + b2\*X42 + b3\*X43 = 213.33 | my4 = 213.33

Дисперсія однорідна

115.56530484806197 0.4883666359311134 1.731481709210302 0.6659545035424239

t1<t\_tab; b1=0

t2<t\_tab; b2=0

t3<t\_tab; b3=0

y1\_hat = 214.72+0.00\*x11+0.00\*x12+0.00\*x13 = 214.72

y2\_hat = 214.72+0.00\*x21+0.00\*x22+0.00\*x23 = 214.72

y3\_hat = 214.72+0.00\*x31+0.00\*x32+0.00\*x33 = 214.72

y4\_hat = 214.72+0.00\*x41+0.00\*x42+0.00\*x43 = 214.72

Рівняння регресії адекватно оригіналу при рівні значимості 0,05