МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Лабораторна робота 4

Проведення трьохфакторного експерименту при використанні рівняння регресії з урахуванням ефекту взаємодії

Гуменюк Святослав
4 курс
ІК-92

Викладач:
Кир'янов Артемій Юрійович

1 Лістинг програми

```
import random as rn
import statistics as stat
import math
import itertools
import numpy as np

x1min = -25
x1max = 75
x2min = 25
x2max = 65
x3min = 25
x3max = 40
xMeanMax = (x1max+x2max+x3max)/3.
xMeanMin = (x1min+x2min+x3min)/3.
ymin = 200 + int(xMeanMin)
ymax = 200 + int(xMeanMax)

def normalized_x(x):
    x0 = [0.5*(max(x[i])+min(x[i])) for i in range(len(x))]
    dx = [x0[i]-min(x[i]) for i in range(len(x))]
    return [[round((x[i][j]-x0[i])/dx[i],2) for j in range(len(x[i]))] \
    for i in range(len(x))]

def sum_of_mult(x):
    return sum([np.prod([*zip(*x)][i]) for i in range(len([*zip(*x)]))])

def B(x1,x2,x3,y):
    meanY = [stat.mean(y[i]) for i in range(len(y))]
    m00 = len(x1)
    m10 = sum(x1)
    m20 = sum(x2)
    m30 = sum(x3)
    m40 = np.dot(x1,x2)
    m50 = np.dot(x1,x3)
    m60 = np.dot(x2,x3)
    m70 = sum_of_mult([x1,x2,x3])

    m01 = m10
    m11 = np.dot(x1,x1)
    m21 = np.dot(x1,x2)
    m31 = np.dot(x1,x3)
    m41 = sum_of_mult([x1,x1,x2])
    m51 = sum_of_mult([x1,x1,x2])
```

```
m61 = m70
m71 = sum_of_mult([x1, x1, x2, x3])

m02 = m20
m12 = m21
m22 = np.dot(x2, x2)
m32 = np.dot(x2, x3)
m42 = sum_of_mult([x1, x2, x2])
m52 = m70
m62 = sum_of_mult([x2, x2, x3])
m72 = sum_of_mult([x1, x2, x2, x3])

m03 = m30
m13 = m31
m23 = m32
m33 = np.dot(x3, x3)
m43 = m70
m53 = sum_of_mult([x1, x3, x3])
m63 = sum_of_mult([x2, x3, x3])
m73 = sum_of_mult([x1, x2, x3, x3])

m04 = m40
m14 = m41
m24 = m42
m34 = m43
m44 = sum_of_mult([x1, x1, x2, x2])
m54 = sum_of_mult([x1, x1, x2, x3])
m64 = sum_of_mult([x1, X2, x2, x3])
m74 = sum_of_mult([x1, x1, x2, x2, x3])

m05 = m50
m15 = m51
m25 = m52
m35 = m53
m45 = m54
m55 = sum_of_mult([x1, x1, x3, x3])
m65 = sum_of_mult([x1, x2, x3, x3])
m75 = sum_of_mult([x1, x1, x2, x3, x3])

m06 = m60
m16 = m61
m26 = m62
m36 = m63
m46 = m64
m56 = m65
m66 = sum_of_mult([x2, x2, x3, x3])
```

```
    m76 = sum_of_mult([x1,x2,x2,x3,x3])

    m07 = m70
    m17 = m71
    m27 = m72
    m37 = m73
    m47 = m74
    m57 = m75
    m67 = m76
    m77 = sum_of_mult([x1,x1,x2,x2,x3,x3])

    M = np.array([\
    [m00,m10,m20,m30,m40,m50,m60,m70],[m01,m10,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]])

    meanY = [stat.mean(y[i]) for i in range(len(y))]
    c = []
    c.append(sum(meanY))
    c.append(np.dot(x1,meanY))
    c.append(np.dot(x2,meanY))
    c.append(np.dot(x3,meanY))
    c.append(sum_of_mult([meanY,x1,x2]))
    c.append(sum_of_mult([meanY,x1,x3]))
    c.append(sum_of_mult([meanY,x2,x3]))
    c.append(sum_of_mult([meanY,x1,x2,x3]))

    return np.linalg.inv(M).dot(c)

def A(x1,x2,x3,y):
    meanY = [stat.mean(y[i]) for i in range(len(y))]
    a0 = stat.mean(meanY)
    a1 = np.dot(meanY,x1)/len(x1)
    a2 = np.dot(meanY,x2)/len(x2)
    a3 = np.dot(meanY,x3)/len(x3)
    a4 = sum_of_mult([x1,x2,meanY])/len(x1)
    a5 = sum_of_mult([x1,x3,meanY])/len(x1)
    a6 = sum_of_mult([x2,x3,meanY])/len(x2)
    a7 = sum_of_mult([x1,x3,x2,meanY])/len(x1)
    return [a0,a1,a2,a3,a4,a5,a5,a7]

def kohren_criteria(y):
    s = [stat.pvariance(y[i]) for i in range(len(y))]
    return max(s)/sum(s)
```

```
def student_criteria(x,y, text=False):
    tCr = 2.120
    meanY = [stat.mean(y[i]) for i in range(len(y))]
    s = [stat.pvariance(y[i]) for i in range(len(y))]
    sb = stat.mean(s)
    sBetaS = sb/len(y[0])/len(s)
    sBetaS = math.sqrt(sBetaS)
    t0 = abs(sum_of_mult([meanY,x[0]])/len(x[0]))/sBetaS
    t1 = abs(sum_of_mult([meanY,x[1]])/len(x[1]))/sBetaS
    t2 = abs(sum_of_mult([meanY,x[2]])/len(x[2]))/sBetaS
    t3 = abs(sum_of_mult([meanY,x[3]])/len(x[3]))/sBetaS
    t4 = abs(sum_of_mult([meanY,x[1],x[2]])/len(x[1]))/sBetaS
    t5 = abs(sum_of_mult([meanY,x[1],x[3]])/len(x[1]))/sBetaS
    t6 = abs(sum_of_mult([meanY,x[2],x[3]])/len(x[2]))/sBetaS
    t7 = abs(sum_of_mult([meanY,x[1],x[2],x[3]])/len(x[1]))/sBetaS

    if text==True:
        print('Student criteria =', *[t0,t1,t2,t3,t4,t5,t6,t7])
    return [1 if t>tCr else 0 for t in [t0,t1,t2,t3,t4,t5,t6,t7]]

def fisher_criteria(y,B,x1,x2,x3,sign,text=False):
    d = 0
    for i in sign:
        d+=i
    meanY = [stat.mean(y[i]) for i in range(len(y))]
    newY = [B[0]*sign[0]+B[1]*sign[1]*x1[i]+B[2]*sign[2]*x2[i]\
    +B[3]*sign[3]*x3[i]+B[4]*sign[4]*x1[i]*x2[i]+B[5]*sign[5]*x1[i]*x3[i]+\
    B[6]*sign[6]*x2[i]*x3[i]+B[7]*sign[7]*x1[i]*x2[i]*x3[i] \
    for i in range(len(x1))]
    diff = [(newY[i]-meanY[i])**2 for i in range(len(newY))]
    try:
        Sad = len(y[0])/(len(y)-d)*sum(diff)
    except ZeroDivisionError:
        return False
    s = [stat.pvariance(y[i]) for i in range(len(y))]
    sb = stat.mean(s)
    if Sad/sb < 4.5:
        if text==True:
            print("Fisher criteria =",Sad/sb)
        return True
    else:
        return False


if __name__ == '__main__':
    X0 = [1 for k in range(8)]
```

```
X1 = [rn.choice(range(x1min,x1max)) for k in range(8)]
X2 = [rn.choice(range(x2min,x2max)) for k in range(8)]
X3 = [rn.choice(range(x3min,x3max)) for k in range(8)]
XN = normalized_x([X1,X2,X3])

Seed = 1
while True:
    rn.seed(Seed)
    m = 3
    Gt = 0.7679
    while True:
        Yy = []
        for i in range(m):
            Yy.append([rn.choice(range(ymin,ymax)) for k in range(8)])
        if kohren_criteria([*zip(*Yy)])<Gt:
            break
        m+=1
    if fisher_criteria([*zip(*Yy)],B(X1,X2,X3,[*zip(*Yy)]),\
    XN[0],XN[1],XN[2],student_criteria([X0,*XN],[*zip(*Yy)])):
        break
    Seed+=1

print("Kohren criteria =",kohren_criteria([*zip(*Yy)]))
print("b_i =",*B(X1,X2,X3,[*zip(*Yy)]))
print("a_i =",*A(*XN,[*zip(*Yy)]))
print(student_criteria([X0,*XN],[*zip(*Yy)],True))
print(fisher_criteria([*zip(*Yy)],B(X1,X2,X3,[*zip(*Yy)]),\
XN[0],XN[1],XN[2],student_criteria([X0,*XN],[*zip(*Yy)]),True))

XN.append([XN[0][i]*XN[1][i] for i in range(len(X1))])
XN.append([XN[0][i]*XN[2][i] for i in range(len(X1))])
XN.append([XN[2][i]*XN[1][i] for i in range(len(X2))])
XN.append([XN[0][i]*XN[1][i]*XN[2][i] for i in range(len(X1))])

for row in zip(*(XN+Yy)):
    print(' & '.join(map(str,row)), end='\\\\\n')
```

2 Результат роботи програми

Нормована матриця планування:

	X_{N1}	X_{N2}	X_{N3}	$X_{N1} \cdot X_{N2}$	$X_{N1} \cdot X_{N3}$	$X_{N2} \cdot X_{N3}$	$X_{N1} \cdot X_{N2} \cdot X_{N3}$	Y_1	Y_2	Y_3
1	1.0	-1.0	1.0	-1.0	1.0	-1.0	-1.0	223	211	212
2	0.58	-0.8	-0.29	-0.46	-0.17	0.23	0.13	244	217	242
3	0.56	0.8	0.57	-0.45	-0.3192	0.46	-0.26	214	241	258
4	0.51	1.0	-0.57	0.51	-0.29	-0.57	-0.29	229	241	231
5	-1.0	-1.0	0.43	1.0	-0.43	-0.43	0.43	250	251	229
6	0.8	0.2	0.57	0.16	0.456	0.11	0.09	223	241	253
7	-0.19	1.0	-1.0	-0.19	0.19	-1.0	0.19	248	226	214
8	-0.33	0.93	1.0	-0.30	-0.33	0.93	-0.30	247	245	213

Далі було знайдено нормовані коефіцієнти рівняння:

$$y = 359.14 - 0.04 \cdot x_{n1} - 2.41 \cdot x_{n2} - 3.23 \cdot x_{n3} - 0.01 \cdot x_{n1} \cdot x_{n2} - 0.01 \cdot x_{n1} \cdot x_{n3} + 0.06 \cdot x_{n2} \cdot x_{n3} + 0.0006 \cdot x_{n1} \cdot x_{n1} \cdot x_{n3}$$

Критерій Кохрена показав наступне значення:

$$G_p = 0.266$$

Вищезгадане значення менше за $G_T = 0.767$, а отже дисперсія однорідна.

Критерій Стьюдента показав наступні значення:

$$t_0 = 737.3,\ t_1 = 64.76,\ t_2 = 107.91,\ t_3 = 156.40,\ t_4 = 57.43,\ t_5 = 0.98,\ t_6 = 108.14,\ t_7 = 84.69$$

Значення $t_5 < 2.306$, тому коефіцієнт рівняння регресії приймаємо незначними при рівні значимості 0.05

Таким чином рівняння регресії має вигляд:

$$y = 359.14 - 0.04 \cdot x_{n1} - 2.41 \cdot x_{n2} - 3.23 \cdot x_{n3} - 0.01 \cdot x_{n1} \cdot x_{n2} + 0.06 \cdot x_{n2} \cdot x_{n3} + 0.0006 \cdot x_{n1} \cdot x_{n1} \cdot x_{n3}$$

Також було знайдено натуралызовані коефіцєнти рівнняння регресії:

$$y = 233.45 + 20.51 \cdot x_{n1} + 34.17 \cdot x_{n2} + 49.52 \cdot x_{n3} - 18.19 \cdot x_{n1} \cdot x_{n2} + 0.312 \cdot x_{n2} \cdot x_{n3} - 26.82 \cdot x_{n1} \cdot x_{n1} \cdot x_{n3}$$

Останнім кроком була перевірка адекватносі моделі за допомогою критерію Фішера:

$$F_p = 0.54$$

Оскільки $F_p < 4.5$, отже рівняння регресії адекватно оригіналу при рівні значимості 0.05

3 Висновки

В ході даної лабораторної роботи було проведено трьохфакторний експеримент з використанням рівняння регресії з урахуванням взаємодії. Було перевірено однорідність десперсії за критерієм Кохрена. Після цього було знайдено натуралізовані коефіцієнти, та визначено значимість коефіцієнтів за допомогою критерію Стьюдента, який показав, що один з коефіцієнтів є незначним. Адекватность рівняння оригіналу було перевірено за допомогою критерію Фішера, який показав, що рівнняння є адекватним оригіналу.