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»

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:
Студент ІІ курсу ФІОТ
Групи ІО-92
Кушенко Сергій

ПЕРЕВІРИВ:
Регіда П.Г.

Київ 2021 р.

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

Варіант завдання:

213	-15	30	25	65	-15	-5

Лістинг програми:

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

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

X_var = ((-15, 30), (25, 65), (-15, -5))
ymax = 200 + sum([i[1] for i in X_var]) / 3  #230
ymin = 200 + sum([i[0] for i in X_var]) / 3  #198.3

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 _ in range(N)] for _ in range(m)]
while True:
    #критерій Кохрена
    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 < cr.G_q005[nearest(cr.forG[0], f2)][nearest(cr.forG[0], f1)] * 10**(-4):
        break
    else:
        m += 1
        print("m = %s,   N = %s,   f1 = %s,   f2 = %s,   Gp = %s,   Gt = %s" % (m, N,
f1, f2, Gp, cr.G_q005[f2-2][f1-1] * 10**(-4)))
        print("Збільшуємо к-сть дослідів\n")
        for i in range(len(Y)):
            Y[i].append(random.randrange(int(ymin), int(ymax)))
```

```
#критерій Стьюдента

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
t_tabl = cr.t[nearest(cr.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)]))


#критерій Фішера

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 = cr.F[nearest(cr.forF[0], f3)][nearest(cr.forF[1], f4)]


print('X:',X)
print('Y:',Y)
print("\nКритерій Корхена")
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("y = %.2f + (%.2f) * x1 + (%.2f) * x2 + (%.2f) * x3" % tuple([round(i, 4) for i
in b]))

print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (b[0], b[1],
X_var[0][0], b[2], X_var[1][0], b[3], X_var[2][0], b[0] + b[1] * X_var[0][0] + b[2] *
X_var[1][0] + b[3] * X_var[2][0]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (b[0], b[1], X_var[0][0], b[3],
X_var[2][1], b[0] + b[1] * X_var[0][0] + b[2] * X_var[1][1] + b[3] * X_var[2][1]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (b[0], b[1], X_var[0][1], b[3],
X_var[2][1], b[0] + b[1] * X_var[0][1] + b[2] * X_var[1][0] + b[3] * X_var[2][1]))
print("%.2f + (%.2f) * (%s) + (%.2f) * (%s) = %.1f" % (b[0], b[1], X_var[0][1], b[3],
X_var[2][0], b[0] + b[1] * X_var[0][1] + b[2] * X_var[1][1] + b[3] * X_var[2][0]))

print("\nКритерій Стьюдента")
print("S:", [round(i, 4) for i in S])
print("Gp:", round(Gp, 4))
print("Gt:", round(cr.G_q005[nearest(cr.forG[0], f2)][nearest(cr.forG[1], f1)]*10**(-
4), 4))
print("S_B:",round(S_B, 4))
print("S_b = %s  sqrt_S_b = %s"%(round(S_b, 4), 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("Табличнне 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("\nКритерій Фішера")
print("d:", d)
print("f4:", f4)
print("f3:", f3)
print("Sад:", Sad)
print("Ft:", Ft)
print("Fp:", Fp)
if Fp > Ft:
    print("Рівняння регресії неадекватне оригіналу (Fp > Ft)")
else:
    print("Рівняння регресії адекватне оригіналу (Fp < Ft)")
```

Результат виконання роботи:

```
X: [[-15, -15, 30, 30], [25, 65, 25, 65], [-15, -5, -5, -15]]
Y: [[218, 217, 211, 200], [213, 211, 221, 218], [201, 199, 228, 215]]

Критерій Корхена
Y сер.: [210.6667, 209.0, 220.0, 211.0]
mx: [7.5, 45.0, -10.0]
my: 212.6667
a: [1658.75, 9516.6667, -2117.5]
aa: [[562.5, 337.5, -75.0], [337.5, 2425.0, -450.0], [-75.0, -450.0, 125.0]]
b: [221.3889, 0.1259, -0.1333, 0.3667]
f1 = 2;  f2 = 4
y = 221.39 + (0.13) * x1 + (-0.13) * x2 + (0.37) * x3
221.39 + (0.13) * (-15) + (-0.13) * (25) + (0.37) * (-15) = 210.7
221.39 + (0.13) * (-15) + (0.37) * (-5) = 209.0
221.39 + (0.13) * (30) + (0.37) * (-5) = 220.0
221.39 + (0.13) * (30) + (0.37) * (-15) = 211.0

Критерій Стьюдента
S: [50.8889, 56.0, 48.6667, 62.0]
Gp: 0.285
Gt: 0.7679
S_B: 54.3889
S_b = 4.5324  sqrt_S_b = 2.1289
beta: [212.6667, 2.8333, -2.6667, 1.8333]
t: [99.893, 1.3309, 1.2526, 0.8611]
f3: 8
Табличнне знач. t: 2.306
tzn: [99.893, 0, 0, 0]
y = 221.39 + (0.00) * x1 + (0.00) * x2 + (0.00) * x3
221.39 + (0.00) * (-15) + (0.00) * (25) + (0.00) * (-15) = 221.4
221.39 + (0.00) * (-15) + (0.00) * (-5) = 221.4
221.39 + (0.00) * (30) + (0.00) * (-5) = 221.4
221.39 + (0.00) * (30) + (0.00) * (-15) = 221.4
```

```
Критерій Фішера
d: 1
f4: 3
f3: 8
Sад: 378.30864197537215
Ft: 4.1
Fp: 83.4674838270484
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, критерій Стьюдента для перевірки нуль-гіпотези та критерій Фішера перевірки адекватності гіпотези. Отримані результати при перевірці є адекватними.

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.