Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

з дисципліни МОПЕ

на тему:

# «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:

студент групи ІВ-82

Гевеленко Назар Романович

Залікова книжка № 8205

Варіант: 204

Перевірив:

Регіда П.Г.

Київ 2020

****

Варіант:

X1min = 15 X2min = 15 X3min = 15

X1max = 45 X2max = 50 X3max = 30

**Код програми:**

import math

import numpy as np

from scipy.stats import t,f

import random as r

from functools import partial

import prettytable as p

m = 3

prob = 0.95

X1\_min = 15

X1\_max = 45

X2\_min = 15

X2\_max = 50

X3\_min = 15

X3\_max = 30

k = 3

X\_ranges = [[X1\_min, X1\_max], [X2\_min, X2\_max], [X3\_min, X3\_max]]

X0\_norm = [1, 1, 1, 1]

X1\_norm = [-1, -1, 1, 1]

X2\_norm = [-1, 1, -1, 1]

X3\_norm = [-1, 1, 1, -1]

N = len(X1\_norm)

Xcp\_max = (X1\_max + X2\_max + X3\_max) / 3

Xcp\_min = (X1\_min + X2\_min + X3\_min) / 3

X\_norm = [X1\_norm, X2\_norm, X3\_norm]

Y\_min = 200 + Xcp\_min

Y\_max = 200 + Xcp\_max

X\_abs = [[max(X\_ranges[j]) if i == 1 else min(X\_ranges[j]) for i in X\_norm[j]] for j in range(k)]

for i in range(len(X\_abs)):

    print("Абсолютні Х{0}: {1}".format(i+1, X\_abs[i]))

def make\_exp(m):

    return [[r.randint(math.floor(Y\_min), math.floor(Y\_max)) for \_ in range(m)] for i in range(N)]

def get\_dispersion(y\_aver, y):

    return sum([(i-y\_aver)\*\*2 for i in y])/len(y)

def get\_average(y):

    return sum(y)/len(y)

def count\_F(a, b):

    return max([a, b])/min([a, b])

def y\_regr\_norm(x1, x2, x3):

        return b0 + x1\*b1 + x2\*b2 + x3\*b3

def y\_regr\_abs(x1, x2, x3):

        return a0 + a1\*x1 + a2\*x2 + a3\*x3

def get\_fisher\_critical(prob,f3, f4):

    for i in [j\*0.001 for j in range(int(10/0.001))]:

        if abs(f.cdf(i,f4,f3)-prob) < 0.0001:

            return i

def get\_student\_critical(prob, f3):

    for i in [j\*0.0001 for j in range(int(5/0.0001))]:

        if abs(t.cdf(i,f3)-(0.5 + prob/0.1\*0.05)) < 0.000005:

            return i

def get\_cohren\_critical(prob, f1, f2):

    f\_crit = f.isf((1-prob)/f2, f1, (f2-1)\*f1)

    return f\_crit/(f\_crit+f2-1)

def is\_significant\_coef(tkr, t):

    return t > tkr

Y\_exp = make\_exp(m)

flag = True

while(flag):

    table1 = p.PrettyTable()

    table1.add\_column("X0", X0\_norm)

    for i in range(k):

        table1.add\_column("X{0}".format(i+1), X\_norm[i])

    for i in range(m):

        table1.add\_column("Y{0}".format(i+1), [j[i] for j in Y\_exp])

    print("Нормалізована матриця:\n", table1)

    mx\_norm\_list = [get\_average(i) for i in X\_norm]

    y\_aver = [get\_average(i) for i in Y\_exp]

    my = get\_average(y\_aver)

    a1 = get\_average([X\_norm[0][i]\*y\_aver[i] for i in range(N)])

    a2 = get\_average([X\_norm[1][i]\*y\_aver[i] for i in range(N)])

    a3 = get\_average([X\_norm[2][i]\*y\_aver[i] for i in range(N)])

    a11 = get\_average([X\_norm[0][i]\*\*2 for i in range(N)])

    a22 = get\_average([X\_norm[1][i]\*\*2 for i in range(N)])

    a33 = get\_average([X\_norm[2][i]\*\*2 for i in range(N)])

    a12 = get\_average([X\_norm[0][i]\*X\_norm[1][i] for i in range(N)])

    a13 = get\_average([X\_norm[0][i]\*X\_norm[2][i] for i in range(N)])

    a23 = get\_average([X\_norm[1][i]\*X\_norm[2][i] for i in range(N)])

    a21 = a12

    a31 = a13

    a32 = a23

    znam = np.array([[1, mx\_norm\_list[0], mx\_norm\_list[1],  mx\_norm\_list[2]],

                      [mx\_norm\_list[0], a11, a12, a13],

                      [mx\_norm\_list[1], a12, a22, a32],

                      [mx\_norm\_list[2], a13, a23, a33]])

    b0\_matr = np.array([[my, mx\_norm\_list[0], mx\_norm\_list[1],  mx\_norm\_list[2]],

                      [a1, a11, a12, a13],

                      [a2, a12, a22, a32],

                      [a3, a13, a23, a33]])

    b1\_matr = np.array([[1, my, mx\_norm\_list[1],  mx\_norm\_list[2]],

                      [mx\_norm\_list[0], a1, a12, a13],

                      [mx\_norm\_list[1], a2, a22, a32],

                      [mx\_norm\_list[2], a3, a23, a33]])

    b2\_matr = np.array([[1, mx\_norm\_list[0], my,  mx\_norm\_list[2]],

                      [mx\_norm\_list[0], a11, a1, a13],

                      [mx\_norm\_list[1], a12, a2, a32],

                      [mx\_norm\_list[2], a13, a3, a33]])

    b3\_matr = np.array([[1, mx\_norm\_list[0], mx\_norm\_list[1],  my],

                      [mx\_norm\_list[0], a11, a12, a1],

                      [mx\_norm\_list[1], a12, a22, a2],

                      [mx\_norm\_list[2], a13, a23, a3]])

    zanm\_value = np.linalg.det(znam)

    b0 = np.linalg.det(b0\_matr)/zanm\_value

    b1 = np.linalg.det(b1\_matr)/zanm\_value

    b2 = np.linalg.det(b2\_matr)/zanm\_value

    b3 = np.linalg.det(b3\_matr)/zanm\_value

    print("Рівняння регресії для нормованих значень:\ny = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3".format(b0, b1, b2, b3))

    print("Підставимо нормованi значення Х в рівння регресії")

    print("Р-ня регресії для Х11, Х21, Х31 =",y\_regr\_norm(X\_norm[0][0], X\_norm[1][0], X\_norm[2][0]))

    print("Середнє y1 =", y\_aver[0])

    print("Р-ня регресії для Х12, Х22, Х32 =", y\_regr\_norm(X\_norm[0][1], X\_norm[1][1], X\_norm[2][1]))

    print("Середнє y2 =", y\_aver[1])

    print("Р-ня регресії для Х13, Х23, Х33 =", y\_regr\_norm(X\_norm[0][2], X\_norm[1][2], X\_norm[2][2]))

    print("Середнє y3 =", y\_aver[2])

    delt\_x1 = (X1\_max - X1\_min)/2

    delt\_x2 = (X2\_max - X2\_min)/2

    delt\_x3 = (X3\_max - X3\_min)/2

    x10 = (X1\_max + X1\_min)/2

    x20 = (X2\_max + X2\_min)/2

    x30 = (X3\_max + X3\_min)/2

    a0 = b0 - b1\*(x10/delt\_x1) - b2\*(x20/delt\_x2) - b3\*(x30/delt\_x3)

    a1 = b1/delt\_x1

    a2 = b2/delt\_x2

    a3 = b3/delt\_x3

    print("Рівняння регресії для абсолютних значень:\ny = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3".format(a0, a1, a2, a3))

    print("Підставимо абсолютні значення Х в рівння регресії")

    print("Р-ня регресії для Х11, Х21, Х31 =",y\_regr\_abs(X\_abs[0][0], X\_abs[1][0], X\_abs[2][0]))

    print("Середнє y1 =", y\_aver[0])

    print("Р-ня регресії для Х12, Х22, Х32 =", y\_regr\_abs(X\_abs[0][1], X\_abs[1][1], X\_abs[2][1]))

    print("Середнє y2 =", y\_aver[1])

    print("Р-ня регресії для Х13, Х23, Х33 =", y\_regr\_abs(X\_abs[0][2], X\_abs[1][2], X\_abs[2][2]))

    print("Середнє y3 =", y\_aver[2])

    print("Р-ня регресії для Х14, Х24, Х34 =", y\_regr\_abs(X\_abs[0][3], X\_abs[1][3], X\_abs[2][3]))

    print("Середнє y3 =", y\_aver[3])

    #Кохрен

    y\_disps = [get\_dispersion(y\_aver[i], Y\_exp[i]) for i in range(N)]

    f1 = m - 1

    f2 = N

    f3 = f2\*f1

    Gp = max(y\_disps)/sum(y\_disps)

    Gkr = get\_cohren\_critical(prob, f1, f2)

    print("--------------------------------------------------------")

    if(Gkr > Gp):

        print("Gkr = {0} > Gp = {1} ---> Дисперсії однорідні".format(Gkr, Gp))

        flag = False

    else:

        print("Gkr = {0} < Gp = {1} ---> Дисперсії неоднорідні, збільшимо m і проведемо розрахунки".format(Gkr, Gp))

        Y\_exp[0].append(r.randint(math.floor(Y\_min), math.floor(Y\_max)))

        Y\_exp[1].append(r.randint(math.floor(Y\_min), math.floor(Y\_max)))

        Y\_exp[2].append(r.randint(math.floor(Y\_min), math.floor(Y\_max)))

        Y\_exp[3].append(r.randint(math.floor(Y\_min), math.floor(Y\_max)))

        m += 1

#Стьюдент

S2B = sum(y\_disps)/N

S2b = S2B/(N\*m)

Sb = math.sqrt(S2b)

beta0 = sum([y\_aver[i]\*X0\_norm[i] for i in range(N)])/N

beta1 = sum([y\_aver[i]\*X1\_norm[i] for i in range(N)])/N

beta2 = sum([y\_aver[i]\*X2\_norm[i] for i in range(N)])/N

beta3 = sum([y\_aver[i]\*X3\_norm[i] for i in range(N)])/N

t0 = abs(beta0)/Sb

t1 = abs(beta1)/Sb

t2 = abs(beta2)/Sb

t3 = abs(beta3)/Sb

tkr = get\_student\_critical(prob, f3)

d = sum([1 if is\_significant\_coef(tkr, i) else 0 for i in [t0, t1, t2, t3]])

a0 = a0 if is\_significant\_coef(tkr, t0) else 0

a1 = a1 if is\_significant\_coef(tkr, t1) else 0

a2 = a2 if is\_significant\_coef(tkr, t2) else 0

a3 = a3 if is\_significant\_coef(tkr, t3) else 0

y\_new = [y\_regr\_abs(X\_abs[0][i], X\_abs[1][i], X\_abs[2][i]) for i in range(N)]

print("--------------------------------------------------------\nПісля перевірки значимості коефіцієнтів: ")

print("Кількість значимих коефіцієнтів:", d)

print("Рівняння регресії для абсолютних значень:\ny = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3".format(a0, a1, a2, a3))

print("Підставимо абсолютні значення Х в рівння регресії")

print("Р-ня регресії для Х11, Х21, Х31 =", y\_new[0])

print("Р-ня регресії для Х12, Х22, Х32 =", y\_new[1])

print("Р-ня регресії для Х13, Х23, Х33 =", y\_new[2])

print("Р-ня регресії для Х14, Х24, Х34 =", y\_new[3])

#Фішер

print("--------------------------------------------------------")

f4 = N - d

S2ad = (m/(N-d))\*sum([(y\_new[i] - y\_aver[i])\*\*2 for i in range(N)])

Fp = S2ad/S2b

Fkr = get\_fisher\_critical(prob, f3, f4)

if(Fkr > Fp):

    print("Fkr = {0} > Fp = {1} ---> Р-ня адекватне оригіналу".format(Fkr, Fp))

else:

    print("Fkr = {0} < Fp = {1} ---> Р-ня неадекватне оригіналу".format(Fkr, Fp))