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-92

Копайло Я.Р.

Залікова книжка № 9213

Варіант: 211

Перевірив:

Регіда П.Г.

Київ 2021

**Варіант:**

****

X1min = 10 X2min = -35 X3min = 10

X1max = 60 X2max = 25 X3max = 15

**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 = **10**x1\_max = **60**x2\_min = -**35**x2\_max = **15**x3\_min = **10**x3\_max = **15**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 = []  
  
for i in range(k):  
 temp = []  
 for j in x\_norm[i]:  
 if j == **1**:  
 temp.append(x\_ranges[i][**1**])  
 else:  
 temp.append(x\_ranges[i][**0**])  
 x\_abs.append(temp)  
print('Абсолютні значення: ' + str(x\_abs))  
  
Y\_exp = []  
for i in range(N):  
 temp = []  
 for \_ in range(m):  
 temp.append(r.randint(math.floor(Y\_min)**,** math.floor(Y\_max)))  
 Y\_exp.append(temp)  
  
  
def y\_perevirka\_norm(x1**,** x2**,** x3):  
 return b0 + x1 \* b1 + x2 \* b2 + x3 \* b3  
  
  
def y\_perevirka\_abs(x1**,** x2**,** x3):  
 return a0 + a1 \* x1 + a2 \* x2 + a3 \* x3  
  
  
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 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  
  
  
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 = [np.mean(i) for i in x\_norm]  
 y\_aver = [np.mean(i) for i in Y\_exp]  
 my = np.mean(y\_aver)  
 a1 = np.mean([x\_norm[**0**][i] \* y\_aver[i] for i in range(N)])  
 a2 = np.mean([x\_norm[**1**][i] \* y\_aver[i] for i in range(N)])  
 a3 = np.mean([x\_norm[**2**][i] \* y\_aver[i] for i in range(N)])  
 a11 = np.mean([x\_norm[**0**][i] \*\* **2** for i in range(N)])  
 a22 = np.mean([x\_norm[**1**][i] \*\* **2** for i in range(N)])  
 a33 = np.mean([x\_norm[**2**][i] \*\* **2** for i in range(N)])  
 a12 = np.mean([x\_norm[**0**][i] \* x\_norm[**1**][i] for i in range(N)])  
 a13 = np.mean([x\_norm[**0**][i] \* x\_norm[**2**][i] for i in range(N)])  
 a23 = np.mean([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]])  
  
 znam\_value = np.linalg.det(znam)  
 b0 = np.linalg.det(b0\_matr) / znam\_value  
 b1 = np.linalg.det(b1\_matr) / znam\_value  
 b2 = np.linalg.det(b2\_matr) / znam\_value  
 b3 = np.linalg.det(b3\_matr) / znam\_value  
 print("Рівняння регресії для нормованих значень:\ny = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3".format(b0**,** b1**,** b2**,** b3))  
 print("Перевірка знайденого рівняння")  
 print("Р-ня регресії для Х11, Х21, Х31 ="**,** y\_perevirka\_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\_perevirka\_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\_perevirka\_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("Перевірка абсолютних значень")  
 print("Р-ня регресії для Х11, Х21, Х31 ="**,** y\_perevirka\_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\_perevirka\_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\_perevirka\_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\_perevirka\_abs(x\_abs[**0**][**3**]**,** x\_abs[**1**][**3**]**,** x\_abs[**2**][**3**]))  
 print("Середнє y3 ="**,** y\_aver[**3**])  
  
 # Кохрен  
 y\_var = [np.var(Y\_exp[i]) for i in range(N)]  
 flag = False  
 f1 = m - **1** f2 = N  
 f3 = f2 \* f1  
 Gp = max(y\_var) / sum(y\_var)  
 Gkr = get\_cohren\_critical(prob**,** f1**,** f2)  
 print('-' \* **100**)  
 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\_var) / 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 tkr < i else **0** for i in [t0**,** t1**,** t2**,** t3]])  
  
a0 = a0 if tkr < t0 else **0**a1 = a1 if tkr < t1 else **0**a2 = a2 if tkr < t2 else **0**a3 = a3 if tkr < t3 else **0**y\_new = [y\_perevirka\_abs(x\_abs[**0**][i]**,** x\_abs[**1**][i]**,** x\_abs[**2**][i]) for i in range(N)]  
  
print("-" \* **100**)  
print("Після перевірки значимості коефіцієнтів: ")  
print("Рівняння регресії для абсолютних значень:\ny = {0} + {1}\*x1 + {2}\*x2 + {3}\*x3".format(a0**,** a1**,** a2**,** a3))  
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("-" \* **100**)  
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))

**Результати роботи:**





