Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2

Тема: «Проведення двофакторного експерименту з

Використанням лінійного рівняння регресії»

Виконав:

студент групи ІО-93

Ващенко Ірина Олегівна

Номер в списку групи: 4

Перевірив:

ас. Регiда П. Г.

Київ 2021

**Лабораторна робота №2**

Тема: «Проведення двофакторного експерименту з

Використанням лінійного рівняння регресії»

**Мета:** провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

**Завдання на лабораторну роботу**

1)Записати лінійне рівняння регресії.  
2)Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору (хо=1).  
3)Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку y). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні ymin ÷ ymax

ymax = (30 - Nваріанту)\*10, ymin = (20 - Nваріанту)\*10.

**Варiант№4:**



import numpy as np  
import math  
  
variant = 4  
m = 5  
x1\_min\_max = [15, 45]  
x2\_min\_max = [30, 80]  
y\_min\_max = [(20 - variant)\*10, (30 - variant)\*10]  
x\_table = [[x1\_min\_max[0], x2\_min\_max[0]],  
 [x1\_min\_max[0], x2\_min\_max[1]],  
 [x1\_min\_max[1], x2\_min\_max[0]]]  
  
Probability = (0.99, 0.98, 0.95, 0.90)  
Rkr = {2: (1.73, 1.72, 1.71, 1.69),  
 6: (2.16, 2.13, 2.10, 2.00),  
 8: (2.43, 4.37, 2.27, 2.17),  
 10: (2.62, 2.54, 2.41, 2.29),  
 12: (2.75, 2.66, 2.52, 2.39),  
 15: (2.9, 2.8, 2.64, 2.49),  
 20: (3.08, 2.96, 2.78, 2.62)}  
  
  
y\_matrix = np.random.randint(\*y\_min\_max, size=(3, m))  
#y\_matrix = [[205, 233, 268, 239, 262],  
# [190, 269, 222, 235, 270],  
# [170, 232, 179, 236, 170]  
# ]  
labels\_table = [**"x1"**, **"x2"**] + [**"y{}"**.format(i+1) for i in range(m)]  
rows\_table = [x\_table[i] + list(y\_matrix[i]) for i in range(3)]  
  
y\_avarg = [sum(y\_matrix[i][j] for j in range(m)) / m for i in range(3)]  
y\_var = np.var(y\_matrix, axis=1)  
sigma\_teta = np.sqrt((2 \* (2 \* m - 2)) / m \* (m - 4))  
  
  
Fuv\_n = [y\_var[0]/y\_var[1], y\_var[2]/y\_var[0], y\_var[2]/y\_var[1]]  
Sigma\_tn = [((m - 2))/m \* Fuv\_n[num] for num in range(3)]  
R\_n = [abs(Sigma\_tn[num] - 1)/sigma\_teta for num in range(3)]  
  
  
def odnoridna\_dispersion():  
 M = min(Rkr, key=lambda x: abs(x - m))  
 p = 0  
 for ruv in (R\_n[0], R\_n[1], R\_n[2]):  
 if ruv > Rkr[M][0]:  
 return **"неоднорідна"** for rkr in range(len(Rkr[M])):  
 if ruv < Rkr[M][rkr]:  
 p = rkr  
 return **f"однорідна p =** {Probability[p]}**"**mx1 = (-1 + 1 - 1) / 3  
mx2 = (-1 - 1 + 1) / 3  
my = sum(y\_avarg) / 3  
  
a1 = (1 + 1 + 1) / 3  
a2 = (1 - 1 - 1) / 3  
a3 = (1 + 1 + 1) / 3  
a11 = (-1 \* y\_avarg[0] + 1 \* y\_avarg[1] - 1 \* y\_avarg[2]) / 3  
a22 = (-1 \* y\_avarg[0] - 1 \* y\_avarg[1] + 1 \* y\_avarg[2]) / 3  
  
B\_n = np.linalg.solve([[1, mx1, mx2],  
 [mx1, a1, a2],  
 [mx2, a2, a3]],  
 [my, a11, a22])  
  
  
def chack\_norm\_equation():  
 x1\_2 = [-1, 1]  
 norm\_equation = [round((B\_n[0] + B\_n[1]\*x1\_2[0] + B\_n[2]\*x1\_2[0]), 2),  
 round((B\_n[0] + B\_n[1]\*x1\_2[1] + B\_n[2]\*x1\_2[0]), 2),  
 round((B\_n[0] + B\_n[1]\*x1\_2[0] + B\_n[2]\*x1\_2[1]), 2)]  
 if norm\_equation[0] != y\_avarg[0] and norm\_equation[1] != y\_avarg[1] and norm\_equation[2] != y\_avarg[2]:  
 print(**"Нормаване рівняння регресії НЕ сходяться"**)  
 return norm\_equation  
  
#Проведемо натуралізацію коефіцієнтів:  
dx1 = math.fabs(x1\_min\_max[1] - x1\_min\_max[0]) / 2  
dx2 = math.fabs(x2\_min\_max[1] - x2\_min\_max[0]) / 2  
  
x10 = (x1\_min\_max[1] + x1\_min\_max[0]) / 2  
x20 = (x2\_min\_max[1] + x2\_min\_max[0]) / 2  
  
a\_0 = B\_n[0] - B\_n[1] \* x10/dx1 - B\_n[2] \* x20/dx2  
a\_1 = B\_n[1]/dx1  
a\_2 = B\_n[2]/dx2  
  
  
#Запишемо натуралізоване рівняння регресії:  
natyraliz\_Y = [round((a\_0 + a\_1\*x1\_min\_max[0] + a\_2\*x2\_min\_max[0]),2),  
 round((a\_0 + a\_1\*x1\_min\_max[1] + a\_2\*x2\_min\_max[0]),2),  
 round((a\_0 + a\_1\*x1\_min\_max[0] + a\_2\*x2\_min\_max[1]),2)]  
  
  
  
print(**"->Нормована матриця<-"**)  
print((**" "**\*4).join(labels\_table))  
print(**"**\n**"**.join([**" "**.join(map(lambda j: **"{:<5}"**.format(j), rows\_table[i])) for i in range(len(rows\_table))]))  
print(**"**\t**"**)  
print(**"σθ:"**, sigma\_teta)  
print(**"Fuv\_n:"**, Fuv\_n)  
print(**"σn:"**, Sigma\_tn)  
print(**"Ruv:"**, R\_n)  
print()  
print(**"Дисперсія"**, odnoridna\_dispersion())  
print()  
print(**"Натуралізовані коефіцієнти"**, [a\_0, a\_1, a\_2])  
print()  
print(**"Середне Y:"**, y\_avarg)  
print(**"Натуралізоване рівняння регресії:"**, natyraliz\_Y)  
print(**"Нормоване рівняння регресії: "**, chack\_norm\_equation())

Table

Description automatically generated with medium confidence

**Контрольні запитання:**

*1)Що таке регресійні поліноми і де вони застосовуються?*

В плануванні експеримента найважливіше є оцінка результатів вимірювань і для цього використовують апроксимуючі поліноми. В ТМП вони називаються регресійні поліноми.

*2)  Визначення однорідності дисперсії.*

Дисперсії реалізацій (вимірювань функції відгуку) σ2j (j=1,N) для нормально розподіленої випадкової величини однакові для усіх комбінації, тобто не залежать від абсолют-ного значення функції відгуку для кожної комбінації. Ця властивість нормально розподіленої випадкової величини

*3)  Що називається повним факторним експериментом?*

Повний факторний експеримент ПФЕ, - це такий факторний експеримент, коли використовуються усі можливі комбінації рівнів факторів. Для ПФЕ кількість комбінацій Nn=rk: якщо r=2, тоді Nn=2k; якщо r=3, тоді Nn=3k; якщо r=5, тоді Nn=5k тощо

**Висновок:**

В процесі лабораторної роботи було проведено двофакторний експеримент, перевірили однорідність дисперсії за критерієм Романовського, отримали коефіцієнти рівняння регресії, а також провели натуралізацію рівняння регресії.