Міністерство освіти і науки України Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського» Факультет інформатики та обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 3

з дисципліни «Методи наукових досліджень»

на тему «ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ.»

ВИКОНАВ:

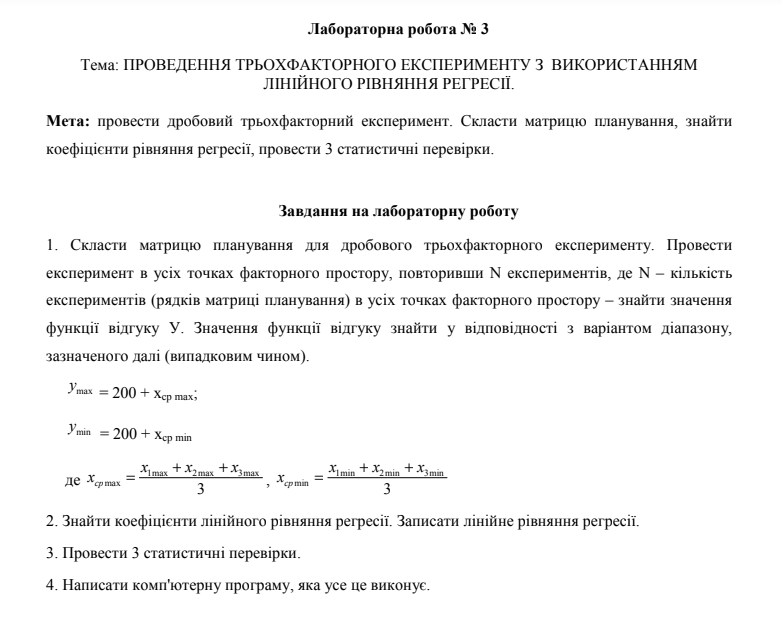
студент 2 курсу групи ІВ-91 Петрук С.В.

Залікова – 9124

ПЕРЕВІРИВ:

ас. Регіда П. Г.

Київ – 2021







**Програмний Код**

from random import randint  
from numpy.linalg import det  
from functools import reduce  
  
  
def naturalize(matrix\_of\_plan, min\_max\_arr):  
 result = []  
 for i in matrix\_of\_plan:  
 result.append(min\_max\_arr[1]) if i == 1 else result.append(min\_max\_arr[0])  
 return result  
  
  
def main():  
 m = 3  
 x1 = [10, 50]  
 x2 = [20, 60]  
 x3 = [20, 25]  
 print(f'x1\_min = {x1[0]}, x1\_max = {x1[1]}')  
 print(f'x2\_min = {x2[0]}, x2\_max = {x2[1]}')  
 print(f'x3\_min = {x3[0]}, x3\_max = {x3[1]}')  
  
 # Матриця планування експерименту з +1,-1  
 x0\_plan1 = [1, 1, 1, 1]  
 x1\_plan1 = [-1, -1, 1, 1]  
 x2\_plan1 = [-1, 1, -1, 1]  
 x3\_plan1 = [-1 \* (x1\_plan1[i] \* x2\_plan1[i]) for i in range(len(x1\_plan1))]  
 print('x0:', x0\_plan1)  
 print('x1:', x1\_plan1)  
 print('x2:', x2\_plan1)  
 print('x3:', x3\_plan1)  
  
 # Матриця планування з натуралізованими значеннями факторів  
 x1\_plan2 = naturalize(x1\_plan1, x1)  
 x2\_plan2 = naturalize(x2\_plan1, x2)  
 x3\_plan2 = naturalize(x3\_plan1, x3)  
 print()  
 print('x1:', x1\_plan2)  
 print('x2:', x2\_plan2)  
 print('x3:', x3\_plan2)  
  
 x\_avg\_max = (max(x1\_plan2) + max(x2\_plan2) + max(x3\_plan2)) / 3  
 x\_avg\_min = (min(x1\_plan2) + min(x2\_plan2) + min(x3\_plan2)) / 3  
 print()  
 print(f'x\_avg\_max = {x\_avg\_max}')  
 print(f'x\_avg\_min = {x\_avg\_min}')  
  
 # Діапазон y  
 y\_max = int(200 + x\_avg\_max)  
 y\_min = int(200 + x\_avg\_min)  
 print()  
 print(f'y\_max = {y\_max}')  
 print(f'y\_min = {y\_min}')  
  
 y1 = [randint(y\_min, y\_max) for \_ in range(4)]  
 y2 = [randint(y\_min, y\_max) for \_ in range(4)]  
 y3 = [randint(y\_min, y\_max) for \_ in range(4)]  
 print('y1:', y1)  
 print('y2:', y2)  
 print('y3:', y3)  
  
 y\_avg\_arr = [(y1[i] + y2[i] + y3[i]) / 3 for i in range(4)]  
 print('y average:', y\_avg\_arr)  
  
 # Математичне очікування  
 mx1 = reduce(lambda a, b: a + b, x1\_plan2) / 4  
 mx2 = reduce(lambda a, b: a + b, x2\_plan2) / 4  
 mx3 = reduce(lambda a, b: a + b, x3\_plan2) / 4  
 my = reduce(lambda a, b: a + b, y\_avg\_arr) / 4  
 print()  
 print(f'mx1 = {mx1}')  
 print(f'mx2 = {mx2}')  
 print(f'mx3 = {mx3}')  
 print(f'my = {my}')  
  
 a1 = sum([x1\_plan2[i] \* y\_avg\_arr[i] for i in range(4)]) / 4  
 a2 = sum([x2\_plan2[i] \* y\_avg\_arr[i] for i in range(4)]) / 4  
 a3 = sum([x3\_plan2[i] \* y\_avg\_arr[i] for i in range(4)]) / 4  
 print()  
 print(f'a1 = {a1}')  
 print(f'a2 = {a2}')  
 print(f'a3 = {a3}')  
  
 a11 = sum([i \* i for i in x1\_plan2]) / 4  
 a22 = sum([i \* i for i in x2\_plan2]) / 4  
 a33 = sum([i \* i for i in x3\_plan2]) / 4  
 print(f'a11 = {a11}')  
 print(f'a22 = {a22}')  
 print(f'a33 = {a33}')  
  
 a12 = sum([x1\_plan2[i] \* x2\_plan2[i] for i in range(4)]) / 4  
 a13 = sum([x1\_plan2[i] \* x3\_plan2[i] for i in range(4)]) / 4  
 a23 = sum([x2\_plan2[i] \* x3\_plan2[i] for i in range(4)]) / 4  
 a21 = a12  
 a31 = a13  
 a32 = a23  
 print(f'a12 = {a12}')  
 print(f'a13 = {a13}')  
 print(f'a23 = {a23}')  
 print(f'a21 = {a21}')  
 print(f'a31 = {a31}')  
 print(f'a32 = {a32}')  
  
 b0 = det([[my, mx1, mx2, mx3],  
 [a1, a11, a12, a13],  
 [a2, a21, a22, a23],  
 [a3, a31, a32, a33]]) / det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a21, a22, a23],  
 [mx3, a31, a32, a33]])  
 b1 = det([[1, my, mx2, mx3],  
 [mx1, a1, a12, a13],  
 [mx2, a2, a22, a23],  
 [mx3, a3, a32, a33]]) / det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a21, a22, a23],  
 [mx3, a31, a32, a33]])  
 b2 = det([[1, mx1, my, mx3],  
 [mx1, a11, a1, a13],  
 [mx2, a21, a2, a23],  
 [mx3, a31, a3, a33]]) / det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a21, a22, a23],  
 [mx3, a31, a32, a33]])  
 b3 = det([[1, mx1, mx2, my],  
 [mx1, a11, a12, a1],  
 [mx2, a21, a22, a2],  
 [mx3, a31, a32, a3]]) / det([[1, mx1, mx2, mx3],  
 [mx1, a11, a12, a13],  
 [mx2, a21, a22, a23],  
 [mx3, a31, a32, a33]])  
  
 print(f'y = {b0} + {b1}\*x1 + {b2}\*x2 + {b3}\*x3')  
  
 for i in range(4):  
 y = b0 + b1 \* x1\_plan2[i] + b2 \* x2\_plan2[i] + b3 \* x3\_plan2[i]  
 print('y =', y)  
  
 # Перевірка однорідності дисперсії за критерієм Кохрена  
 dispersion = [((y1[i] - y\_avg\_arr[i]) \*\* 2 + (y2[i] - y\_avg\_arr[i]) \*\* 2 + (y3[i] - y\_avg\_arr[i]) \*\* 2) / 3 for i in range(4)]  
 print('dispersion:', dispersion)  
  
 gp = max(dispersion) / sum(dispersion)  
 print('Gp =', gp)  
  
 # Рівень значимості q = 0.05; f1 = m - 1 = 2; f2 = N = 4  
 # За таблицею Gт = 0.7679  
 if gp < 0.7679:  
 print('Дисперсія однорідна')  
 else:  
 print('Дисперсія неоднорідна')  
 exit()  
  
 # Оцінка значимості коефіцієнтів регресії згідно критерію Стьюдента  
 s2b = sum(dispersion) / 4  
 s2bs\_avg = s2b/4\*m  
 sb = s2bs\_avg \*\* (1/2)  
  
 beta0 = sum([y\_avg\_arr[i] \* x0\_plan1[i] for i in range(4)]) / 4  
 beta1 = sum([y\_avg\_arr[i] \* x1\_plan1[i] for i in range(4)]) / 4  
 beta2 = sum([y\_avg\_arr[i] \* x2\_plan1[i] for i in range(4)]) / 4  
 beta3 = sum([y\_avg\_arr[i] \* x3\_plan1[i] for i in range(4)]) / 4  
  
 beta\_arr = [beta0, beta1, beta2, beta3]  
 print('beta:', beta\_arr)  
 t\_arr = [abs(beta\_arr[i])/sb for i in range(4)]  
 print('t:', t\_arr)  
  
 # f3 = f1\*f2 = 2\*4 = 8  
 # З таблиці беремо значення 2.306  
 indexes = []  
 for i, v in enumerate(t\_arr):  
 if t\_arr[i] > 2.306:  
 indexes.append(i)  
 else:  
 print(f'Коефіцієнт b{i} = {v} приймаємо не значним')  
  
 b\_list = [b0, b1, b2, b3]  
 print(f'y = b{indexes[0]}')  
  
 b\_res = [b\_list[indexes[0]] for \_ in range(4)]  
 for i in b\_res:  
 print(f'y = {i}')  
  
 # Критерій Фішера  
 # кількість значимих коефіцієнтів  
 d = 1  
 s2\_ad = m \* sum([(y\_avg\_arr[i] - b\_res[i])\*\*2 for i in range(4)]) / 4 - d  
 fp = s2\_ad/s2b  
 print(f'Fp = {fp}')  
  
 # Fт = 4.5  
 if fp > 4.5:  
 print('Рівняння регресії неадекватно оригіналу при рівні значимості 0.05')  
 else:  
 print('Рівняння регресії адекватно оригіналу при рівні значимості 0.05')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Результат програми

