Міністерство освіти і науки України

Національний технічний університет України «КПІ» імені Ігоря Сікорського

Кафедра обчислювальної техніки ФІОТ

ЗВІТ

з лабораторної роботи №2

з навчальної дисципліни «Методи наукових досліджень»

Тема:

ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

Виконав:

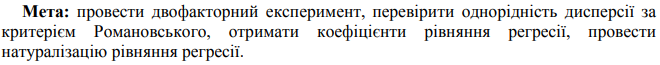
Студент групи ІВ-92,

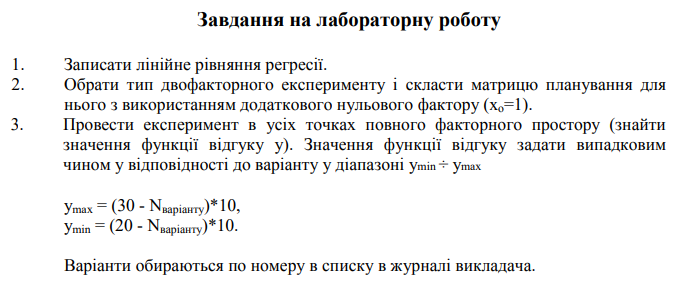
Карпека Дмитрій Юрійович

Перевірив:

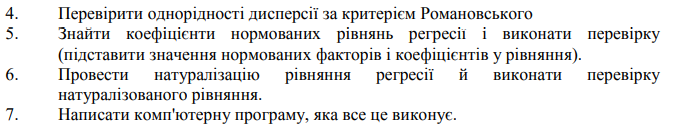
Регіда П. Г.

Київ 2021



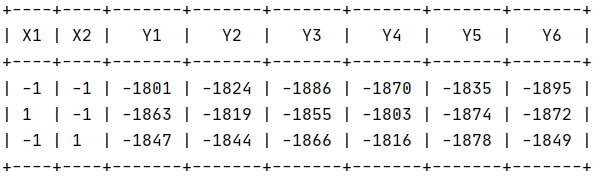




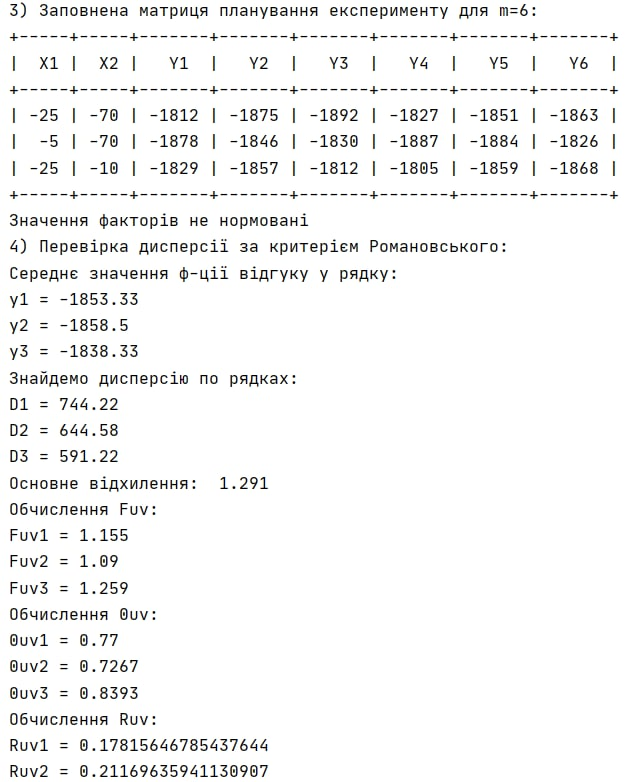


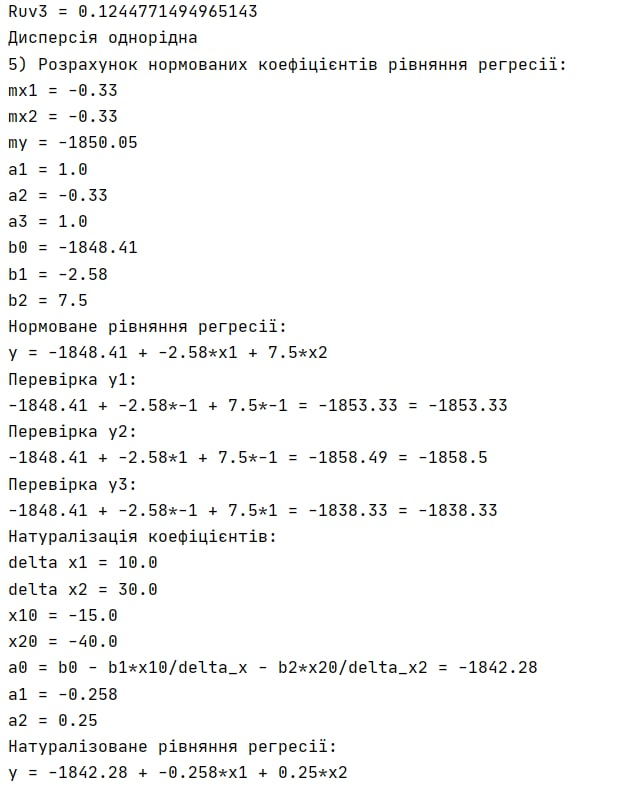
Виконання роботи:

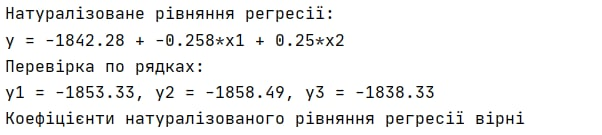
1. Нормована матриця планування повного факторного експерименту:



1. Результати роботи програми:







Код програми:

from random import randint  
import prettytable  
import math  
import numpy  
*# Лабораторна робота №2 "ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З  
# ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ" з предмету МОПЕ  
# Варіянт №210 Карпека Дмитрій  
  
#--------------------------------------------------Початкові умови-----------------------------------------------------*variant = 210  
  
y\_max = (30 - variant) \* 10  
y\_min = (20 - variant) \* 10  
  
*#--------------------------------------------------1-------------------------------------------------------------------  
# y = b0 + b1X1 + b2X2  
#--------------------------------------------------2-------------------------------------------------------------------*x0 = 1  
x1\_min = -25  
x1\_min\_norm = -1  
x1\_max = -5  
x1\_max\_norm = 1  
x2\_min = -70  
x2\_min\_norm = -1  
x2\_max = -10  
x2\_max\_norm = 1  
m = 6  
  
pt = prettytable.PrettyTable()  
pt.field\_names = [**"X1"**, **"X2"**] + [**"Y"** + str(x) for x in range(1, m+1)]  
  
*#--------------------------------------------------3-------------------------------------------------------------------*def matrix\_plan(m, ymin, ymax, n=3):  
 return [[randint(ymin, ymax) for \_ in range(m)] for \_ in range(n)]  
  
  
matrix\_y = matrix\_plan(m, y\_min, y\_max)  
matrix\_y = [[x1\_min\_norm, x2\_min\_norm] + matrix\_y[0],  
 [x1\_max\_norm, x2\_min\_norm] + matrix\_y[1],  
 [x1\_min\_norm, x2\_max\_norm] + matrix\_y[2]]  
pt.add\_rows(matrix\_y)  
  
print(**"3) Заповнена матриця планування експерименту для m={0}:"**.format(m))  
print(pt)  
print(**"Значення факторів нормовані"**)  
*#--------------------------------------------------4-------------------------------------------------------------------  
#4.1*def average(list):  
 return sum(list) / len(list)  
  
  
y1 = round(average(matrix\_y[0][2:len(matrix\_y[0])]), 2)  
y2 = round(average(matrix\_y[1][2:len(matrix\_y[1])]), 2)  
y3 = round(average(matrix\_y[2][2:len(matrix\_y[2])]), 2)  
print(**"4) Перевірка дисперсії за критерієм Романовського:"**)  
print(**"Середнє значення ф-ції відгуку у рядку:**\n**y1 = {0}**\n**y2 = {1}**\n**y3 = {2}"**.format(y1, y2, y3))  
*#4.2*def disp(row\_y, y\_average):  
 sigma\_squared = 0  
 for y in row\_y:  
 sigma\_squared += (y - y\_average)\*\*2  
 sigma\_squared = sigma\_squared/len(row\_y)  
 return sigma\_squared  
  
  
d1 = round(disp(matrix\_y[0][2:len(matrix\_y[0])], y1), 2)  
d2 = round(disp(matrix\_y[1][2:len(matrix\_y[1])], y2), 2)  
d3 = round(disp(matrix\_y[2][2:len(matrix\_y[2])], y3), 2)  
print(**"Знайдемо дисперсію по рядках:**\n**D1 = {0}**\n**D2 = {1}**\n**D3 = {2}"**.format(d1, d2, d3))  
*#4.3*funddev = round(math.sqrt((2\*(2\*m - 2))/(m\*(m-4))), 3)  
print(**"Основне відхилення: "**, funddev)  
*#4.4*def fuvn(dn, dm):  
 if dn >= dm:  
 fuv = dn/dm  
 else:  
 fuv = dm/dn  
 return fuv  
  
  
fuv1 = round(fuvn(d1, d2), 3)  
fuv2 = round(fuvn(d2, d3), 3)  
fuv3 = round(fuvn(d1, d3), 3)  
print(**"Обчислення Fuv:**\n**Fuv1 = {0}**\n**Fuv2 = {1}**\n**Fuv3 = {2}"**.format(fuv1, fuv2, fuv3))  
*#4.5*def tetta\_uvn(fuv, m=m):  
 return ((m - 2)/m \* fuv)  
  
  
tetta\_uv1 = round(tetta\_uvn(fuv1), 4)  
tetta\_uv2 = round(tetta\_uvn(fuv2), 4)  
tetta\_uv3 = round(tetta\_uvn(fuv3), 4)  
print(**"Обчислення 0uv:**\n**0uv1 = {0}**\n**0uv2 = {1}**\n**0uv3 = {2}"**.format(tetta\_uv1, tetta\_uv2, tetta\_uv3))  
*#4.6*def ruvn(tetta, fundev=funddev):  
 return abs(tetta - 1)/fundev  
  
ruv1 = ruvn(tetta\_uv1)  
ruv2 = ruvn(tetta\_uv2)  
ruv3 = ruvn(tetta\_uv3)  
print(**"Обчислення Ruv:**\n**Ruv1 = {0}**\n**Ruv2 = {1}**\n**Ruv3 = {2}"**.format(ruv1, ruv2, ruv3))  
*#4.7*def criteria(r1, r2, r3, probability, m=m):  
 table = [[0, 2, 6, 8, 10, 12, 15, 20],  
 [0.99, 1.73, 2.16, 2.43, 2.62, 2.75, 2.9, 3.08],  
 [0.98, 1.72, 2.13, 2.37, 2.54, 2.66, 2.8, 2.96],  
 [0.95, 1.71, 2.1, 2.27, 2.41, 2.52, 2.64, 2.78],  
 [0.9, 1.69, 2, 2.17, 2.29, 2.39, 2.49, 2.62]]  
 rkr = 0  
 for i in table[1:len(table)]:  
 if i[0] == probability:  
 rkr = i[table[0].index(m)]  
 if r1 < rkr and r2 < rkr and r3 < rkr:  
 print(**"Дисперсія однорідна"**)  
 else:  
 print(**"Дисперсія неоднорідна"**)  
  
criteria(ruv1, ruv2, ruv3, 0.9)  
*#5*x1min\_n, x2min\_n = -1, -1  
x1max\_n, x2max\_n = 1, 1  
matrix\_norm = [[x1min\_n, x2min\_n],  
 [x1max\_n, x2min\_n],  
 [x1min\_n, x2max\_n]]  
  
mx1 = sum([i[0] for i in matrix\_norm])/3  
mx2 = sum([i[1] for i in matrix\_norm])/3  
my = sum([y1, y2, y3])/3  
a1 = sum(i[0]\*\*2 for i in matrix\_norm) / 3  
a2 = sum([i[0]\*i[1] for i in matrix\_norm])/3  
a3 = sum(i[1]\*\*2 for i in matrix\_norm) / 3  
  
a11 = sum([matrix\_norm[0][0]\*y1, matrix\_norm[1][0]\*y2, matrix\_norm[2][0]\*y3])/3  
a22 = sum([matrix\_norm[0][1]\*y1, matrix\_norm[1][1]\*y2, matrix\_norm[2][1]\*y3])/3  
  
b0 = round(numpy.linalg.det([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]]) / numpy.linalg.det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]]), 2)  
b1 = round(numpy.linalg.det([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]]) / numpy.linalg.det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]]), 2)  
b2 = round(numpy.linalg.det([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]]) / numpy.linalg.det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]]), 2)  
print(**"5) Розрахунок нормованих коефіцієнтів рівняння регресії:**\n**mx1 = {0}**\n**mx2 = {1}**\n**my = {2}**\n**a1 = {3}**\n**a2 = {4}**\n**a3 = {5}**\n**b0 = {6}**\n**b1 = {7}**\n**b2 = {8}"**.format(round(mx1, 2), round(mx2, 2), round(my, 2), round(a1, 2), round(a2, 2), round(a3, 2), round(b0, 2), round(b1, 2), round(b2, 2)))  
print(**"Нормоване рівняння регресії:**\n**y = {0} + {1}\*x1 + {2}\*x2"**.format(b0, b1, b2))  
  
y1\_check = round(b0 + b1\*matrix\_norm[0][0] + b2\*matrix\_norm[0][1], 2)  
y2\_check = round(b0 + b1\*matrix\_norm[1][0] + b2\*matrix\_norm[1][1], 2)  
y3\_check = round(b0 + b1\*matrix\_norm[2][0] + b2\*matrix\_norm[2][1], 2)  
print(**"Перевірка y1:**\n**{0} + {1}\*{2} + {3}\*{4} = {5} = {6}"**.format(b0, b1, matrix\_norm[0][0], b2, matrix\_norm[0][1], y1\_check, y1))  
print(**"Перевірка y2:**\n**{0} + {1}\*{2} + {3}\*{4} = {5} = {6}"**.format(b0, b1, matrix\_norm[1][0], b2, matrix\_norm[1][1], y2\_check, y2))  
print(**"Перевірка y3:**\n**{0} + {1}\*{2} + {3}\*{4} = {5} = {6}"**.format(b0, b1, matrix\_norm[2][0], b2, matrix\_norm[2][1], y3\_check, y3))  
  
*#6  
#TODO: end 6) and commit lab, pin a link in GT, notify teacher*print(**"Натуралізація коефіцієнтів:"**)  
delta\_x1 = abs(x1\_max - x1\_min)/2  
delta\_x2 = abs(x2\_max - x2\_min)/2  
print(**"delta x1 = {0}**\n**delta x2 = {1}"**.format(delta\_x1, delta\_x2))  
x10 = (x1\_max + x1\_min)/2  
x20 = (x2\_max + x2\_min)/2  
print(**"x10 = {0}**\n**x20 = {1}"**.format(x10, x20))  
a0 = round(b0 - b1\*x10/delta\_x1 - b2\*x20/delta\_x2, 2)  
print(**"a0 = b0 - b1\*x10/delta\_x - b2\*x20/delta\_x2 ="**, a0)  
a1 = round(b1/delta\_x1, 3)  
a2 = round(b2/delta\_x2, 3)  
print(**"a1 = {0}**\n**a2 = {1}"**.format(a1, a2))  
print(**"Натуралізоване рівняння регресії:**\n**y = {0} + {1}\*x1 + {2}\*x2"**.format(a0, a1, a2))  
  
ych1 = round(a0 + a1\*x1\_min + a2\*x2\_min, 2)  
ych2 = round(a0 + a1\*x1\_max + a2\*x2\_min, 2)  
ych3 = round(a0 + a1\*x1\_min + a2\*x2\_max, 2)  
print(**"Перевірка по рядках:**\n**y1 = {0}, y2 = {1}, y3 = {2}"**.format(ych1, ych2, ych3))  
if abs(y1\_check - ych1)/ych1 < 0.05 and abs(y2\_check - ych2)/ych2 < 0.05 and abs(y3\_check - ych3)/ych3 < 0.05:  
 print(**"Коефіцієнти натуралізованого рівняння регресії вірні"**)