Міністерство освіти і науки України

Національний технічний університет України «КПІ» імені Ігоря Сікорського

Кафедра обчислювальної техніки ФІОТ

ЗВІТ

з лабораторної роботи №3

з навчальної дисципліни «Методи наукових досліджень»

Тема:

ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

Виконав:

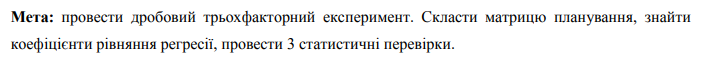
Студент групи ІВ-92,

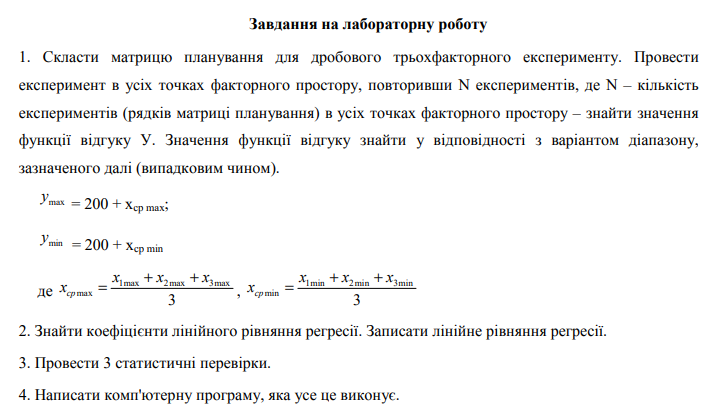
Карпека Дмитрій Юрійович

Перевірив:

Регіда П. Г.

Київ 2021



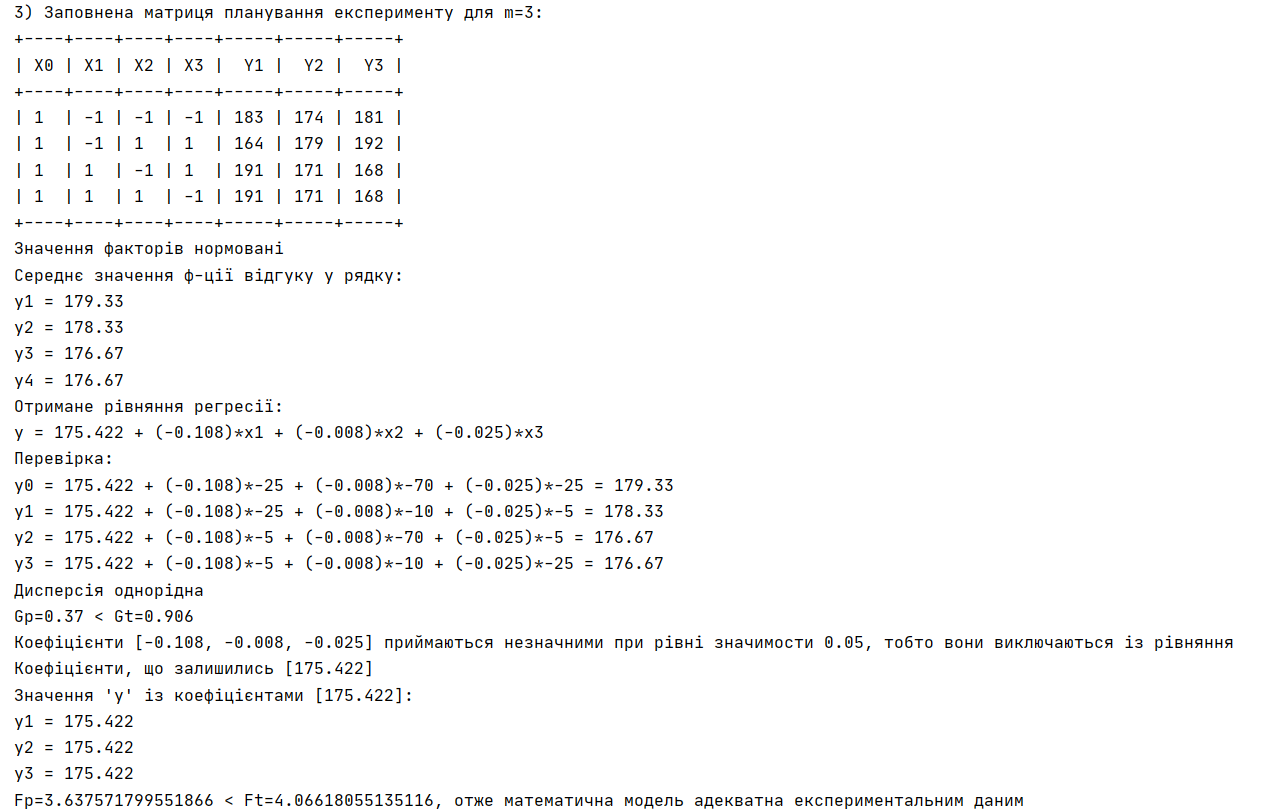


Варіант:



Виконання роботи:

1. Результати роботи програми:



Код програми:

from random import randint  
import prettytable  
import math  
import numpy as np  
from scipy.stats import f, t  
from functools import partial  
import math  
  
*# Лабораторна робота №3 "ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ  
# ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ" з предмету МОПЕ  
# Варіянт №210 Карпека Дмитрій  
  
#--------------------------------------------------Початкові умови-----------------------------------------------------*variant = 210  
*#--------------------------------------------------1-------------------------------------------------------------------  
# y = b0 + b1X1 + b2X2  
#--------------------------------------------------2-------------------------------------------------------------------*x0 = 1  
  
x1\_min = -25  
x1\_max = -5  
x2\_min = -70  
x2\_max = -10  
x3\_min = -25  
x3\_max = -5  
  
x1\_min\_norm, x2\_min\_norm, x3\_min\_norm = -1, -1, -1  
x1\_max\_norm, x2\_max\_norm, x3\_max\_norm = 1, 1, 1  
  
m = 3  
n = 4  
  
x\_average\_max = sum([x1\_max, x2\_max, x3\_max])/3  
x\_average\_min = sum([x1\_min, x2\_min, x3\_min])/3  
  
y\_max = round(200 + x\_average\_max)  
y\_min = round(200 + x\_average\_min)  
  
  
pt = prettytable.PrettyTable()  
pt.field\_names = [**"X0"**, **"X1"**, **"X2"**, **"X3"**] + [**"Y"** + str(x) for x in range(1, m+1)]  
  
  
*#--------------------------------------------------3-------------------------------------------------------------------*def matrix\_plan(m, ymin, ymax, n=n):  
 return [[randint(ymin, ymax) for \_ in range(m)] for \_ in range(n)]  
  
  
matrix\_y = matrix\_plan(m, y\_min, y\_max)  
matrix\_y = [[x0, x1\_min\_norm, x2\_min\_norm, x3\_min\_norm] + matrix\_y[0],  
 [x0, x1\_min\_norm, x2\_max\_norm, x3\_max\_norm] + matrix\_y[1],  
 [x0, x1\_max\_norm, x2\_min\_norm, x3\_max\_norm] + matrix\_y[2],  
 [x0, x1\_max\_norm, x2\_max\_norm, x3\_min\_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][m+1:len(matrix\_y[0])]), 2)  
y2 = round(average(matrix\_y[1][m+1:len(matrix\_y[1])]), 2)  
y3 = round(average(matrix\_y[2][m+1:len(matrix\_y[2])]), 2)  
y4 = round(average(matrix\_y[3][m+1:len(matrix\_y[3])]), 2)  
print(**"Середнє значення ф-ції відгуку у рядку:"**)  
for \_ in range(0, m+1):  
 yn = round(average(matrix\_y[\_][m+1:len(matrix\_y[\_])]), 2)  
 print(**"y{0} = {1}"**.format(\_+1, yn))  
  
matrix\_natur = [[x1\_min, x2\_min, x3\_min],  
 [x1\_min, x2\_max, x3\_max],  
 [x1\_max, x2\_min, x3\_max],  
 [x1\_max, x2\_max, x3\_min]]  
*# TODO: mxn, my, an DONE*mx1 = average([i[0] for i in matrix\_natur])  
mx2 = average([i[1] for i in matrix\_natur])  
mx3 = average([i[2] for i in matrix\_natur])  
my = average([y1, y2, y3, y4])  
  
a1 = average([matrix\_natur[0][0]\*y1, matrix\_natur[1][0]\*y2, matrix\_natur[2][0]\*y3, matrix\_natur[3][0]\*y4])  
a2 = average([matrix\_natur[0][1]\*y1, matrix\_natur[1][1]\*y2, matrix\_natur[2][1]\*y3, matrix\_natur[3][1]\*y4])  
a3 = average([matrix\_natur[0][2]\*y1, matrix\_natur[1][2]\*y2, matrix\_natur[2][2]\*y3, matrix\_natur[3][2]\*y4])  
*# TODO: aij, i==j DONE*a11 = average([x\*x for x in [matrix\_natur[0][0], matrix\_natur[1][0], matrix\_natur[2][0], matrix\_natur[3][0]]])  
a22 = average([x\*x for x in [matrix\_natur[0][1], matrix\_natur[1][1], matrix\_natur[2][1], matrix\_natur[3][1]]])  
a33 = average([x\*x for x in [matrix\_natur[0][2], matrix\_natur[1][2], matrix\_natur[2][2], matrix\_natur[3][2]]])  
*# TODO: aij DONE*a12 = average([matrix\_natur[0][0]\*matrix\_natur[0][1], matrix\_natur[1][0]\*matrix\_natur[1][1], matrix\_natur[2][0]\*matrix\_natur[2][1], matrix\_natur[3][0]\*matrix\_natur[3][1]])  
a21 = a12  
a13 = average([matrix\_natur[0][0]\*matrix\_natur[0][2], matrix\_natur[1][0]\*matrix\_natur[1][2], matrix\_natur[2][0]\*matrix\_natur[2][2], matrix\_natur[3][0]\*matrix\_natur[3][2]])  
a31 = a13  
a23 = average([matrix\_natur[0][1]\*matrix\_natur[0][2], matrix\_natur[1][1]\*matrix\_natur[1][2], matrix\_natur[2][1]\*matrix\_natur[2][2], matrix\_natur[3][1]\*matrix\_natur[3][2]])  
a32 = a23  
*# TODO: bi DONE*b0 = np.linalg.det(np.array([[my, mx1, mx2, mx3], [a1, a11, a12, a13], [a2, a12, a22, a32], [a3, a13, a23, a33]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))  
b1 = np.linalg.det(np.array([[1, my, mx2, mx3], [mx1, a1, a12, a13], [mx2, a2, a22, a32], [mx3, a3, a23, a33]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))  
b2 = np.linalg.det(np.array([[1, mx1, my, mx3], [mx1, a11, a1, a13], [mx2, a12, a2, a32], [mx3, a13, a3, a33]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))  
b3 = np.linalg.det(np.array([[1, mx1, mx2, my], [mx1, a11, a12, a1], [mx2, a12, a22, a2], [mx3, a13, a23, a3]])) / np.linalg.det(np.array([[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]))  
print(**"Отримане рівняння регресії:**\n**y = {0} + ({1})\*x1 + ({2})\*x2 + ({3})\*x3"**.format(round(b0, 3), round(b1, 3), round(b2, 3), round(b3, 3)))  
*# TODO: check DONE*print(**"Перевірка:"**)  
for i in range(4):  
 y = b0 + b1\*matrix\_natur[i][0] + b2\*matrix\_natur[i][1] + b3\*matrix\_natur[i][2]  
 print(**"y{0} = {1} + ({2})\*{3} + ({4})\*{5} + ({6})\*{7} = {8}"**.format(i, round(b0,3), round(b1,3), matrix\_natur[i][0], round(b2,3), matrix\_natur[i][1], round(b3,3), matrix\_natur[i][2], round(y,2)))  
  
*# TODO Cohren criteria DONE*s\_sq\_y1 = average([(y1j - y1)\*\*2 for y1j in matrix\_y[0][m+1:len(matrix\_y[0])]])  
s\_sq\_y2 = average([(y2j - y2)\*\*2 for y2j in matrix\_y[1][m+1:len(matrix\_y[1])]])  
s\_sq\_y3 = average([(y3j - y3)\*\*2 for y3j in matrix\_y[2][m+1:len(matrix\_y[2])]])  
s\_sq\_y4 = average([(y4j - y4)\*\*2 for y4j in matrix\_y[3][m+1:len(matrix\_y[3])]])  
Gp = max([s\_sq\_y1, s\_sq\_y2, s\_sq\_y3, s\_sq\_y4])/sum([s\_sq\_y1, s\_sq\_y2, s\_sq\_y3, s\_sq\_y4])  
  
f1 = m-1  
f2 = n  
f3 = f1\*f2  
q = 0.05  
  
def cohren(f1, f2, q=q):  
 q1 = q / f1  
 fisher\_value = f.ppf(q=1 - q1, dfn=f2, dfd=(f1 - 1) \* f2)  
 return fisher\_value / (fisher\_value + f1 - 1)  
Gt = cohren(f1, f2)  
if Gp < Gt:  
 print(**"Дисперсія однорідна"**)  
 print(**f"Gp=**{round(Gp,3)} **< Gt=**{round(Gt,3)}**"**)  
else:  
 print(**"Дисперсія неоднорідна"**)  
  
*# Student criteria DONE*s\_sq\_aver = average([s\_sq\_y1, s\_sq\_y2, s\_sq\_y3, s\_sq\_y4])  
s\_sq\_b = s\_sq\_aver/(n\*m)  
s\_b = math.sqrt(s\_sq\_b)  
  
student = partial(t.ppf, q=1 - 0.025)  
t\_student = student(df=f3)  
  
bet0 = sum(list(map(lambda x, y: x\*y, [y1, y2, y3, y4], [row[0] for row in matrix\_y])))/n  
bet1 = sum(list(map(lambda x, y: x\*y, [y1, y2, y3, y4], [row[1] for row in matrix\_y])))/n  
bet2 = sum(list(map(lambda x, y: x\*y, [y1, y2, y3, y4], [row[2] for row in matrix\_y])))/n  
bet3 = sum(list(map(lambda x, y: x\*y, [y1, y2, y3, y4], [row[3] for row in matrix\_y])))/n  
  
t0 = abs(bet0)/s\_b  
t1 = abs(bet1)/s\_b  
t2 = abs(bet2)/s\_b  
t3 = abs(bet3)/s\_b  
  
ts = [t0, t1, t2, t3]  
res\_c = [x[1] for x in list(map(lambda t, b: [t, b], [t0, t1, t2, t3], [b0, b1, b2, b3])) if x[0] > t\_student]  
res\_t = [x[0] for x in list(map(lambda t, b: [t, b], [t0, t1, t2, t3], [b0, b1, b2, b3])) if x[0] > t\_student]  
excluded\_c = [x[1] for x in list(map(lambda t, b: [t, b], [t0, t1, t2, t3], [b0, b1, b2, b3])) if not x[0] > t\_student]  
  
for i in range(4):  
 matrix\_natur[i].insert(0, 1)  
def result\_x(ni):  
 res\_x = []  
 for t in res\_t:  
 if ts.index(t) == res\_t.index(t):  
 res\_x.append(matrix\_natur[ni][ts.index(t)])  
 return res\_x  
  
print(**f"Коефіцієнти** {[round(c,3) for c in excluded\_c]} **приймаються незначними при рівні значимости 0.05, тобто вони виключаються із рівняння"**)  
print(**"Коефіцієнти, що залишились "** + str([round(c,3) for c in res\_c]))  
  
def regression(b, x):  
 return round(sum(list(map(lambda i, j: i\*j, b, x))),3)  
  
print(**f"Значення 'y' із коефіцієнтами** {[round(c,3) for c in res\_c]}**:"**)  
for i in range(1, 4):  
 print(**"y{0} = {1}"**.format(i, regression(res\_c, result\_x(i-1))))  
  
*# Fisher criteria*y1\_f = regression(res\_c, result\_x(0))  
y2\_f = regression(res\_c, result\_x(1))  
y3\_f = regression(res\_c, result\_x(2))  
y4\_f = regression(res\_c, result\_x(3))  
d = len(res\_c)  
f4 = n - d  
s\_sq\_ad = (m/(n - d)) \* sum(list(map(lambda y, yf: (y - yf)\*\*2, [y1, y2, y3, y4], [y1\_f, y2\_f, y3\_f, y4\_f])))  
fp = s\_sq\_ad/s\_sq\_b  
  
fisher = partial(f.ppf, q=1 - 0.05)  
f\_t = fisher(dfn=f4, dfd=f3)  
if fp < f\_t:  
 print(**f'Fp=**{fp} **< Ft=**{f\_t}**, отже математична модель адекватна експериментальним даним'**)  
else:  
 print(**f'Fp=**{fp} **>= Ft=**{f\_t}**, отже математична модель не адекватна експериментальним даним'**)