Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЛАБОРАТОРНА РОБОТА № 3
з дисципліни «Методи оптимізації та планування експерименту»
на тему «Проведення трьохфакторного експерименту з використанням лінійного рівняння регресії»

ВИКОНАВ:
Студент II курсу
ФІОТ
Групи ІО-93
Варіант №18
Май Т. Н.

ПЕРЕВІРИВ:

Регіда П.Г.

Київ 2021 р.

Мета: провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

Завдання на лабораторну роботу

1. Скласти матрицю планування для дробового трьохфакторного експерименту. Провести експеримент в усіх точках факторного простору, повторивши N експериментів, де N – кількість експериментів (рядків матриці планування) в усіх точках факторного простору – знайти значення функції відгуку У. Значення функції відгуку знайти у відповідності з варіантом діапазону, зазначеного далі (випадковим чином).

$$y_{max} = 200 + x_{cp\ max};$$

$$y_{min} = 200 + x_{cp\ min}$$

$$\text{де } x_{cp\ max} = \frac{x_{1max} + x_{2max} + x_{3max}}{3},\quad x_{cp\ min} = \frac{x_{1min} + x_{2min} + x_{3min}}{3}$$

2. Знайти коефіцієнти лінійного рівняння регресії. Записати лінійне рівняння регресії.
3. Провести 3 статистичні перевірки.
4. Написати комп'ютерну програму, яка усе це виконує.

Варіант:

318	20	70	-15	45	20	35

Програмний код:

```
from random import *
import numpy as np
from numpy.linalg import solve
from scipy.stats import f, t
from functools import partial

#Варіант 318

# Проведення експерименту з трифакторним пострілом
class Fractional:
    def_init_(self, n, m):
        self.n = n
        self.m = m
        self.x_range = [[20, 70], [-15, 45], [20, 35]]
```

```python
        self.x_min = (20 - 15 + 20) / 3
        self.x_max = (70 + 45 + 35) / 3

        self.y_max = round(200 + self.x_max)
        self.y_min = round(200 + self.x_min)
        # матриця планування ПФЕ
        self.x_n = [[1, -1, -1, -1],
                    [1, -1, 1, 1],
                    [1, 1, -1, 1],
                    [1, 1, 1, -1],
                    [1, -1, -1, 1],
                    [1, -1, 1, -1],
                    [1, 1, -1, -1],
                    [1, 1, 1, 1]]

        self.y = np.zeros(shape=(self.n, self.m))
        self.y_new_values = []
        for i in range(self.n):
            for j in range(self.m):
                self.y[i][j] = randint(self.y_min, self.y_max)
        # середнє значення у
        self.y_av = [round(sum(i) / len(i), 2) for i in self.y]

        self.x_n = self.x_n[:len(self.y)]
        self.x = np.ones(shape=(len(self.x_n), len(self.x_n[0])))

        for i in range(len(self.x_n)):
            for j in range(1, len(self.x_n[i])):
                if self.x_n[i][j] == -1:
                    self.x[i][j] = self.x_range[j - 1][0]
                else:
                    self.x[i][j] = self.x_range[j - 1][1]
        self.f1 = m - 1
        self.f2 = n
        self.f3 = self.f1 * self.f2
        self.q = 0.05

# підстановка у регресію
    def regres(self, x, b):
        y = sum([x[i] * b[i] for i in range(len(x))])
        return y

# Розрахунок коефіцієнтів рівняння регресії
    def count_koefitients(self):
        mx = [(sum(self.x[:, 1]) / self.n),
              (sum(self.x[:, 2]) / self.n),
              (sum(self.x[:, 3]) / self.n)]
        my = sum(self.y_av) / self.n
```

```
        a = [(sum([self.x[i][1] * self.x[i][2] for i in range(len(self.x))]) / se
lf.n),
               (sum([self.x[i][1] * self.x[i][3] for i in range(len(self.x))]) /
self.n),
               (sum([self.x[i][2] * self.x[i][3] for i in range(len(self.x))]) /
self.n),
               (sum([i ** 2 for i in self.x[:, 1]]) / self.n),
               (sum([i ** 2 for i in self.x[:, 2]]) / self.n),
               (sum([i ** 2 for i in self.x[:, 3]]) / self.n)]

        a1 = sum([self.y_av[i] * self.x[i][1] for i in range(len(self.x))]) / sel
f.n
        a2 = sum([self.y_av[i] * self.x[i][2] for i in range(len(self.x))]) / sel
f.n
        a3 = sum([self.y_av[i] * self.x[i][3] for i in range(len(self.x))]) / sel
f.n

        X = [[1, mx[0], mx[1], mx[2]], [mx[0], a[3], a[0], a[1]], [mx[1], a[0], a
[4], a[2]], [mx[2], a[1], a[2], a[5]]]
        Y = [my, a1, a2, a3]
        B = [round(i, 2) for i in solve(X, Y)]
        print('\nРівняння регресії')
        print(f'y = {B[0]} + {B[1]}*x1 + {B[2]}*x2 + {B[3]}*x3')

        return B
# Розрахунок дисперсії
    def dispersion(self):
        res = []
        for i in range(self.n):
            s = sum([(self.y_av[i] - self.y[i][j]) ** 2 for j in range(self.m)])
/ self.m
            res.append(s)
        return res
# Перевірка за критерієм Кохрена
    def kohren(self):
        q1 = self.q / self.f1
        fisher_value = f.ppf(q=1 - q1, dfn=self.f2, dfd=(self.f1 - 1) * self.f2)
        G_cr = fisher_value / (fisher_value + self.f1 - 1)
        s = self.dispersion()
        Gp = max(s) / sum(s)
        return Gp, G_cr

    def student(self):
        # Перевірка за критерієм Стьюдента
        def bs():
            res = [sum(1 * y for y in self.y_av) / self.n]
            for i in range(3):  # 4 - ксть факторів
                b = sum(j[0] * j[1] for j in zip(self.x[:, i], self.y_av)) / self
.n
                res.append(b)
            return res
```

```
        S_kv = self.dispersion()
        s_kv_aver = sum(S_kv) / self.n

        # статистична оцінка дисперсії
        s_Bs = (s_kv_aver / self.n / self.m) ** 0.5
        Bs = bs()
        ts = [abs(B) / s_Bs for B in Bs]
        return ts
# Перевірка адекватності за критерієм Фішера
    def fisher(self, d):
        S_ad = self.m / (self.n - d) * sum([(self.y_new_values[i] - self.y_av[i])
 ** 2 for i in range(len(self.y))])
        S_kv = self.dispersion()
        S_kv_aver = sum(S_kv) / self.n
        F_p = S_ad / S_kv_aver
        return F_p

    def check(self):
        # Проведення статистичних перевірок
        student = partial(t.ppf, q=1 - 0.025)
        t_student = student(df=self.f3)

        print('\nПеревірка за критерієм Кохрена')
        Gp, G_kr = self.kohren()
        print(f'Gp = {Gp}')
        if Gp < G_kr:
            print(f'З ймовірністю {1-self.q} дисперсії однорідні.')
        else:
            print("Необхідно збільшити кількість дослідів")
            self.m += 1
            Fractional(self.n, self.m)

        ts = self.student()
        print('\nПеревірка значущості коефіцієнтів за критерієм Стьюдента')
        print('Критерій Стьюдента:\n', ts)
        res = [t for t in ts if t > t_student]
        B = self.count_koefitients()
        final_k = [B[ts.index(i)] for i in ts if i in res]
        print('Коефіцієнти {} статистично незначущі, тому ми виключаємо їх з рівн
яння.'.format(
            [i for i in B if i not in final_k]))

        for j in range(self.n):
            self.y_new_values.append(self.regres([self.x[j][ts.index(i)] for i in
 ts if i in res], final_k))

        print(f'\nЗначення "y" з коефіцієнтами {final_k}')
        print(self.y_new_values)

        d = len(res)
```

```
        f4 = self.n - d
        F_p = self.fisher(d)

        fisher = partial(f.ppf, q=1 - 0.05)
        f_t = fisher(dfn=f4, dfd=self.f3)
        print('\nПеревірка адекватності за критерієм Фішера')
        print('Fp =', F_p)
        print('F_t =', f_t)
        if F_p < f_t:
            print('Математична модель адекватна експериментальним даним')
        else:
            print('Математична модель не адекватна експериментальним даним')


experiment = Fractional(7, 8)
experiment.check()
```

Вивід програми:

```
Gp = 0.18452221881388067
З ймовірністю 0.95 дисперсії однорідні.

Перевірка значущості коефіцієнтів за критерієм Стьюдента
Критерій Стьюдента:
 [148.17100939545935, 148.17100939545935, 6141.717301718841, 1572.651643896295]

Рівняння регресії
y = 229.85 + 0.0*x1 + -0.03*x2 + -0.04*x3
Коефіцієнти [0.0] статистично незначущі, тому ми виключаємо їх з рівняння.

Значення "y" з коефіцієнтами [229.85, 229.85, -0.03, -0.04]
[459.34999999999997, 456.95, 458.75, 457.54999999999995, 458.75, 457.54999999999995, 459.34999999999997]

Перевірка адекватності за критерієм Фішера
Fp = 7417.219813871622
F_t = 2.7939488515842408
Математична модель не адекватна експериментальним даним
PS C:\Users\maxma\PycharmProjects\snippetproject-basic-master>
```

Контрольні питання:

1) ДФЕ – дробовий факторний експеримент,- це коли використовуються частка ПФЕ, яка кратна степеню двійки, тобто Nд=2-1Nп (напіврепліка), Nд=2-2Nп (1/4 репліки) тощо.

2) Якщо **N≥3** (**k≥2**),-- тоді використовується критерій Кохрена:

1. Серед знайдених статистичних оцінок дисперсії $\mathbf{S^2_j}$ ($j=\overline{1,N}$) знаходять оцінку з максимальним значенням $\mathbf{S^2_{max}=Max\{S^2_j}$ ($j=\overline{1,N}$)$\}$.
2. Розраховують значення критерію Кохрена $\mathbf{G=S^2_{max}/\sum_1^N S^2_j}$
3. Визначають числа ступенів свободи $\mathbf{f_1}$ та $\mathbf{f_2}$: $\mathbf{f_1=m-1}$; $\mathbf{f_2=N}$.
4. Обирають рівень значущості **q**.
5. По спеціальним таблицям Кохрена знаходять критичне (табличне) значення крите-рія Кохрена $\mathbf{G_{кр}}$, яке відповідає значенням **q**, $\mathbf{f_1}$ та $\mathbf{f_2}$.
6. Порівнюють значення **G** та $\mathbf{G_{кр}}$.

Якщо $\mathbf{G \le G_{кр}}$, -- тоді вважається, що гіпотеза стосовно однорідності дисперсії підтвер-джується з ймовірністю **p** (**p=1-q**) і ми можемо виконувати розрахунок коефіцієнтів рівняння регресії.
Якщо $\mathbf{G \ge G_{кр}}$, -- тоді вважається, що гіпотеза стосовно однорідності дисперсії не під-тверджується з ймовірністю **p** (**p=1-q**), -- тоді необхідно збільшити кількість повторів, провести додаткові дослідження і знову здійснити перевірку однорідності дисперсії.

Підтвердження гіпотези стосовно однорідності дисперсії дозволяє отримати більш точну статистичну оцінку дисперсії функції відгуку $\mathbf{S^2=(1/N)\sum_1^N S^2_j}$.

3) Перевірка значущості коефіцієнтів лінійної регресії виконується з допомогою **t**-критерія Стьюдента окремо для кожного коефіцієнта $\mathbf{b_к}$ ($к=\overline{0,k}$), де k кількість факторів, наступним чином:

1.Знаходять значення статистичної оцінки дисперсії помилки при визначенні будь-якого коефіцієнту рівняння регресії **$S^2\{b_к\}$** ($к=\overline{0,k}$). При цьому вважається, що статистичні оцінки дисперсії помилки однакові для усіх коефіцієнтів і розраховуються по наступній формулі: **$S^2\{b_к\}=(1/Nm)S^2$** ($к=\overline{0,k}$), де **S^2** -- статистична оцінка дисперсії функції відгуку (див. статистичну перевірку однорідності дисперсії з використанням критерія Кохрена).
2.Розраховують значення **t**-критеріїв Стьюдента **$t_к$** ($к=\overline{0,k}$) для кожного коефіцієнта рівняння регресії **$b_к$** (к=0,k), використовуючи модулі абсолютних значень коефіцієнтів **$|b_к|$** ($к=\overline{0,k}$) та значення статистичної оцінки дисперсії помилки **$S^2\{b_к\}$** ($к=\overline{0,k}$) по формулі: **$t_к=|b_к|/S^2\{b_к\}$** ($к=\overline{0,k}$).

Перевірка значущості коефіцієнтів нелінійних регресій також виконується з допомогою того ж **t**-критерія Стьюдента окремо для кожного коефіцієнта, але статистична оцінка дисперсії помилки для кожного коефіцієнта рівняння регресії (пункт 1 виконання **t**-критерія) визначається окремою формулою і залежить від композиційного плану, який використовується.

4) Отримане рівняння регресії необхідно перевірити на адекватність досліджуваному об'єкту. Для цієї мети необхідно оцінити, наскільки відрізняються середні значення у вихідної величини, отриманої в точках факторного простору, і значення у, отриманого з рівняння регресії в тих самих точках факторного простору. Для цього використовують дисперсію адекватності. Адекватність моделі перевіряють за F-критерієм Фішера, який дорівнює відношенню дисперсії адекватності до дисперсії відтворюваності:

$F_p = S^2_{ад} / S^2_в$,

де:

$S^2_{ад} = \frac{m}{N-d}\sum_{j=1}^{N}[\hat{y}_j - \overline{y_j}]^2$, де d –кількість значущих коефіцієнтів.

$\hat{y}_j$ – значення функції відгуку при підстановці X_j та отриманих коефіцієнтів b_i у рівняння регресії

$\overline{y_j}$ – середнє значення функції відгуку

$$S^2_B = \sum_{j=1}^{N}\frac{S^2\{y_j\}}{N}$$

$$S^2\{y_j\} = \frac{1}{m}\sum_{g=1}^{m}\left(\overline{y}_j - y_{jg}\right)^2, \text{ де } y_{jk}\ \left(j=\overline{1,N}\right)\left(g=\overline{1,m}\right), \text{ де } \overline{y}_j = \frac{1}{m}\sum_{g=1}^{m} y_{jg}\ \left(j=\overline{1,N}\right)\left(g=\overline{1,m}\right)$$

m – кількість дослідів; кількість вимірів у за однією й тією ж самою комбінації факторів

N – кількість експериментів (рядків матриці планування)

Знайдене шляхом розрахунку F_p порівнюють з табличним значенням $F_т$, що визначається при рівні значимості q та кількості ступенів свободи f4 = N - d і

f3 = f1 * f2

Якщо $F_p < F_т$ то отримана математична модель з прийнятим рівнем статистичної значимості q адекватна експериментальним даним.