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»

з теми «Проведення трьохфакторного експерименту з використанням лінійного рівняння регресії»

Виконав:

студент ІІ курсу ФІОТ

групи ІВ-92

Залога Андрій

Варіант: 208

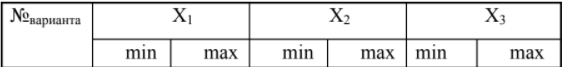
Перевірив:

Регіда П. Г.

Київ 2021

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

**Варіант**

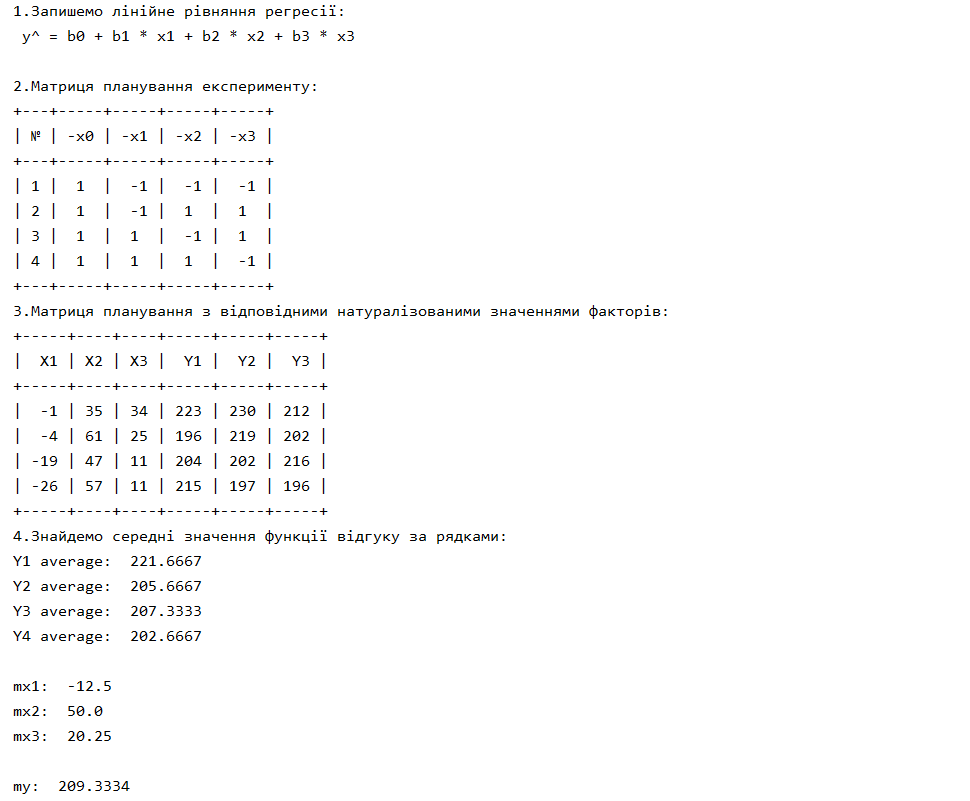
****

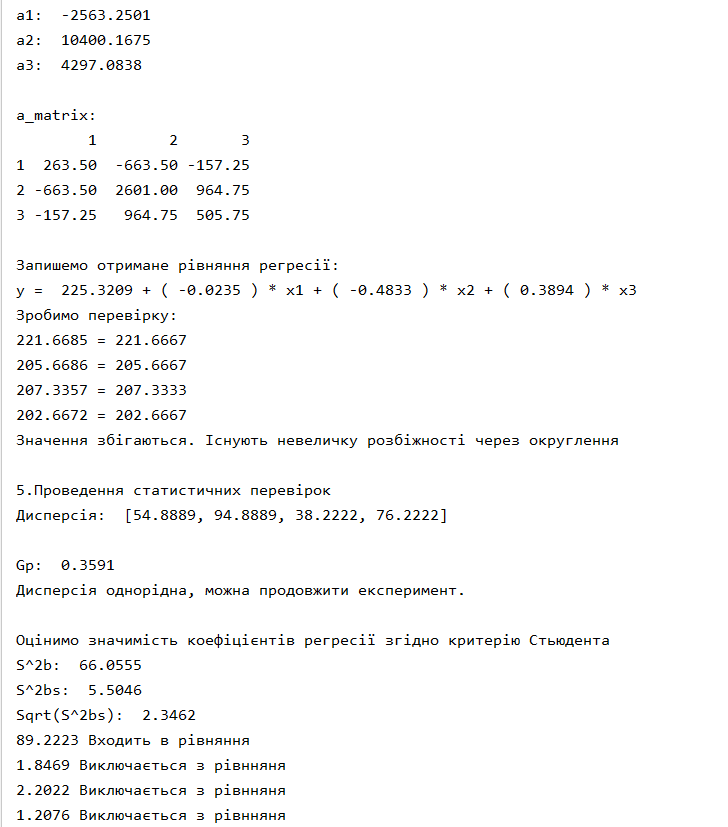
C:\Users\Andrew\Documents\ShareX\Screenshots\2021-03\chrome_FXfu0iuQpg.png

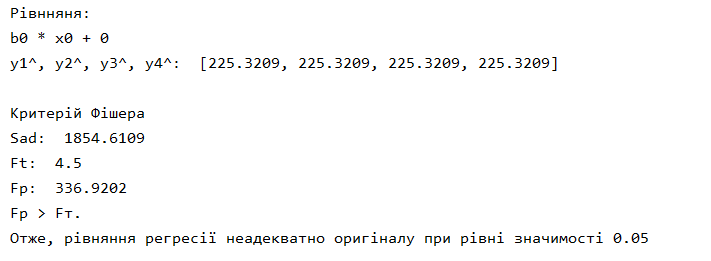
**Програмний код**

**import** numpy **as** np  
**import** random  
**from** prettytable **import** PrettyTable  
**import** pandas **as** pd  
**import** sys  
  
*# 1 пункт*print(**"1.Запишемо лінійне рівняння регресії: \n y^ = b0 + b1 \* x1 + b2 \* x2 + b3 \* x3"**)  
print()  
  
*# 2 пунктт*plan\_matrix = [[1, -1, -1, -1],  
 [1, -1, 1, 1],  
 [1, 1, -1, 1],  
 [1, 1, 1, -1]]  
  
print(**"2.Матриця планування експерименту:"**)  
table1 = PrettyTable()  
table1.field\_names = [**"№"**, **"-x0"**, **"-x1"**, **"-x2"**, **"-x3"**]  
table1.add\_rows([  
 [**"1"**, plan\_matrix[0][0], plan\_matrix[0][1], plan\_matrix[0][2], plan\_matrix[0][3]],  
 [**"2"**, plan\_matrix[1][0], plan\_matrix[1][1], plan\_matrix[1][2], plan\_matrix[1][3]],  
 [**"3"**, plan\_matrix[2][0], plan\_matrix[2][1], plan\_matrix[2][2], plan\_matrix[2][3]],  
 [**"4"**, plan\_matrix[3][0], plan\_matrix[3][1], plan\_matrix[3][2], plan\_matrix[3][3]]  
])  
print(table1)  
  
*# 3 пункт*x1\_min = -30  
x1\_max = 0  
x2\_min = 10  
x2\_max = 60  
x3\_min = 10  
x3\_max = 35  
y\_min, y\_max = int(200 + (x1\_min + x2\_min + x3\_min) / 3), int(200 + (x1\_max + x2\_max + x3\_max) / 3)  
  
  
**def** random\_numbers(x1, x2):  
 array = []  
 **for** i **in** range(4):  
 array.append(random.randint(x1, x2 + 1))  
 **return** array  
  
  
x0 = [1, 1, 1, 1]  
x1 = random\_numbers(x1\_min, x1\_max)  
x2 = random\_numbers(x2\_min, x2\_max)  
x3 = random\_numbers(x3\_min, x3\_max)  
y1 = random\_numbers(y\_min, y\_max)  
y2 = random\_numbers(y\_min, y\_max)  
y3 = random\_numbers(y\_min, y\_max)  
  
print(**"3.Матриця планування з відповідними натуралізованими значеннями факторів:"**)  
table2 = PrettyTable()  
table2.field\_names = [**"X1"**, **"X2"**, **"X3"**, **"Y1"**, **"Y2"**, **"Y3"**]  
**for** i **in** range(len(x1)):  
 table2.add\_row([x1[i], x2[i], x3[i], y1[i], y2[i], y3[i]])  
print(table2)  
  
*# 4 пункт*print(**"4.Знайдемо середні значення функції відгуку за рядками:"**)  
  
y\_average = []  
**for** i **in** range(len(y1)):  
 y\_average.append(round((y1[i] + y2[i] + y3[i]) / 3, 4))  
print(**"Y1 average: "**, y\_average[0])  
print(**"Y2 average: "**, y\_average[1])  
print(**"Y3 average: "**, y\_average[2])  
print(**"Y4 average: "**, y\_average[3])  
print()  
  
mx\_array = [round(sum(x1) / 4, 1), round(sum(x2) / 4, 1), round(sum(x3) / 4, 4)]  
print(**"mx1: "**, mx\_array[0])  
print(**"mx2: "**, mx\_array[1])  
print(**"mx3: "**, mx\_array[2])  
print()  
  
my = round(np.average(y\_average), 4)  
print(**"my: "**, my)  
print()  
  
a\_array = []  
**for** i **in** range(3):  
 a = 0  
 **for** j **in** range(4):  
 a += globals()[**"x"** + str(i + 1)][j] \* y\_average[j]  
 a\_array.append(round(a / 4, 4))  
print(**"a1: "**, a\_array[0])  
print(**"a2: "**, a\_array[1])  
print(**"a3: "**, a\_array[2])  
print()  
  
aa\_array = []  
**for** i **in** range(3):  
 a = 0  
 **for** j **in** range(4):  
 a += globals()[**"x"** + str(i + 1)][j] \*\* 2  
 aa\_array.append(round(a / 4, 4))  
  
a12 = 0  
a13 = 0  
a23 = 0  
**for** j **in** range(4):  
 a12 += x1[j] \* x2[j]  
**for** j **in** range(4):  
 a13 += x1[j] \* x3[j]  
**for** j **in** range(4):  
 a23 += x2[j] \* x3[j]  
  
a12 = a21 = round(a12 / 4, 4)  
a13 = a31 = round(a13 / 4, 4)  
a23 = a32 = round(a23 / 4, 4)  
  
a\_matrix = [[aa\_array[0], a12, a13],  
 [a21, aa\_array[1], a23],  
 [a31, a32, aa\_array[2]]]  
  
print(**"a\_matrix: "**)  
print(pd.DataFrame(a\_matrix, columns=list(**'123'**), index=list(**'123'**)))  
print()  
  
b0 = round(np.linalg.det([[my, mx\_array[0], mx\_array[1], mx\_array[2]],  
 [a\_array[0], aa\_array[0], a12, a13],  
 [a\_array[1], a12, aa\_array[1], a32],  
 [a\_array[2], a13, a23, aa\_array[2]]]) / np.linalg.det(  
 [[1, mx\_array[0], mx\_array[1], mx\_array[2]],  
 [mx\_array[0], aa\_array[0], a12, a13],  
 [mx\_array[1], a12, aa\_array[1], a32],  
 [mx\_array[2], a13, a23, aa\_array[2]]]), 4)  
  
b1 = round(np.linalg.det([[1, my, mx\_array[1], mx\_array[2]],  
 [mx\_array[0], a\_array[0], a12, a13],  
 [mx\_array[1], a\_array[1], aa\_array[1], a32],  
 [mx\_array[2], a\_array[2], a23, aa\_array[2]]]) / np.linalg.det(  
 [[1, mx\_array[0], mx\_array[1], mx\_array[2]],  
 [mx\_array[0], aa\_array[0], a12, a13],  
 [mx\_array[1], a12, aa\_array[1], a32],  
 [mx\_array[2], a13, a23, aa\_array[2]]]), 4)  
  
b2 = round(np.linalg.det([[1, mx\_array[0], my, mx\_array[2]],  
 [mx\_array[0], aa\_array[0], a\_array[0], a13],  
 [mx\_array[1], a12, a\_array[1], a32],  
 [mx\_array[2], a13, a\_array[2], aa\_array[2]]]) / np.linalg.det(  
 [[1, mx\_array[0], mx\_array[1], mx\_array[2]],  
 [mx\_array[0], aa\_array[0], a12, a13],  
 [mx\_array[1], a12, aa\_array[1], a32],  
 [mx\_array[2], a13, a23, aa\_array[2]]]), 4)  
  
b3 = round(np.linalg.det([[1, mx\_array[0], mx\_array[1], my],  
 [mx\_array[0], aa\_array[0], a12, a\_array[0]],  
 [mx\_array[1], a12, aa\_array[1], a\_array[1]],  
 [mx\_array[2], a13, a23, a\_array[2]]]) / np.linalg.det(  
 [[1, mx\_array[0], mx\_array[1], mx\_array[2]],  
 [mx\_array[0], aa\_array[0], a12, a13],  
 [mx\_array[1], a12, aa\_array[1], a32],  
 [mx\_array[2], a13, a23, aa\_array[2]]]), 4)  
print(**"Запишемо отримане рівняння регресії:"**)  
print(**"у = "**, b0, **"+ ("**, b1, **") \* x1 + ("**, b2, **") \* x2 + ("**, b3, **") \* x3"**)  
  
  
**def** regression(x1, x2, x3):  
 **global** b0, b1, b2, b3  
 y = b0 + b1 \* x1 + b2 \* x2 + b3 \* x3  
 **return** round(y, 4)  
  
  
print(**"Зробимо перевірку:"**)  
**for** i **in** range(len(y\_average)):  
 print(regression(x1[i], x2[i], x3[i]), **"="**, y\_average[i])  
print(**"Значення збігаються. Існують невеличку розбіжності через округлення"**)  
print()  
  
*# 5 пункт*print(**"5.Проведення статистичних перевірок"**)  
s\_array = []  
**for** i **in** range(4):  
 s = 0  
 **for** j **in** range(3):  
 s += (globals()[**"y"** + str(j + 1)][i] - y\_average[i]) \*\* 2  
 s\_array.append(round(s / 3, 4))  
print(**"Дисперсія: "**, s\_array)  
print()  
  
gp = round(max(s\_array) / sum(s\_array), 4)  
print(**"Gp: "**, gp)  
  
**if** gp < 0.7679:  
 print(**"Дисперсія однорідна, можна продовжити експеримент."**)  
**else**:  
 print(**"Неоднорідна дисперсія, повторіть експеримент."**)  
 sys.exit()  
print()  
  
print(**"Оцінимо значимість коефіцієнтів регресії згідно критерію Стьюдента"**)  
s\_b = round(sum(s\_array) / 4, 4)  
s\_bs = round(s\_b / 12, 4)  
sqrt\_sbs = round(np.sqrt(s\_bs), 4)  
print(**"S^2b: "**, s\_b)  
print(**"S^2bs: "**, s\_bs)  
print(**"Sqrt(S^2bs): "**, sqrt\_sbs)  
  
beta = []  
teta = []  
  
**for** i **in** range(4):  
 b = 0  
 **for** j **in** range(4):  
 b += y\_average[j] \* plan\_matrix[j][i]  
 beta.append(round(b / 4, 4))  
  
**for** i **in** range(4):  
 teta.append(round(abs(beta[i]) / sqrt\_sbs, 4))  
  
**for** i **in** range(len(teta)):  
 **if** teta[i] > 2.306:  
 print(teta[i], **"Входить в рівняння"**)  
 **else**:  
 print(teta[i], **"Виключається з рівнняня"**)  
  
koef\_t = []  
new\_t = []  
**for** i **in** range(len(teta)):  
 **if** teta[i] > 2.306:  
 koef\_t.append(i)  
  
**for** i **in** range(4):  
 t = 0  
 **for** j **in** koef\_t:  
 t += globals()[**"b"** + str(j)] \* globals()[**"x"** + str(j)][i]  
 new\_t.append(t)  
print(**"Рівнняня:"**)  
  
**for** i **in** koef\_t:  
 print(**"b"** + str(i), **"\*"**, **"x"** + str(i), end=**" "**)  
 print(**"+"**, end=**" "**)  
print(0)  
print(**"y1^, y2^, y3^, y4^: "**, new\_t)  
print()  
  
print(**"Критерій Фішера"**)  
sad = 0  
**for** i **in** range(4):  
 sad += (new\_t[i] - y\_average[i]) \*\* 2  
s\_ad = round(sad \* 1.5, 4)  
fp = round(s\_ad / s\_bs, 4)  
ft = 4.5  
print(**"Sad: "**, s\_ad)  
print(**"Ft: "**, ft)  
print(**"Fp: "**, fp)  
**if** fp > ft:  
 print(**"Fp > Fт. \nОтже, рівняння регресії неадекватно оригіналу при рівні значимості 0.05"**)

**Результати**

****

****

****

**Відповіді на контрольні запитання**

1. **Що називається дробовим факторним експериментом?**  Дробовий факторний експеримент – це частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови лінійної моделі.
2. **Для чого потрібно розрахункове значення Кохрена?** Розрахункове значення Кохрена показує, яку частку в загальній сумі дисперсій у рядках має максимальна з них.
3. **Для чого перевіряється критерій Стьюдента?** Критерій Стьюдента використовується для перевірки значущості коефіцієнтів.
4. **Чим визначається критерій Фішера і як його застосовувати?**  Критерій Фішера використовується для перевірки адекватності рівняння регресії.

**Висновки**

Проведено дробовий трьохфакторний експеримент. Скласдено матрицю планування, знайдені коефіцієнти рівняння регресії, проведені 3 статистичні перевірки. Застосовані критерій Фішера, критерій Стьюдента, розрахункове значення Кохрена.