Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Методи оптимізації та планування експерименту**

Лабораторна робота №3:

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ

ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:

студент групи ІО-83

Коломієць Дмитро Юрійович

Залікова книжка № 8313

Перевірив Регіда П. Г.

Київ 2020р.

**Лабораторна робота №3**

**Тема:** ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ

ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти

коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

**Виконання:**

Варіант – 312.

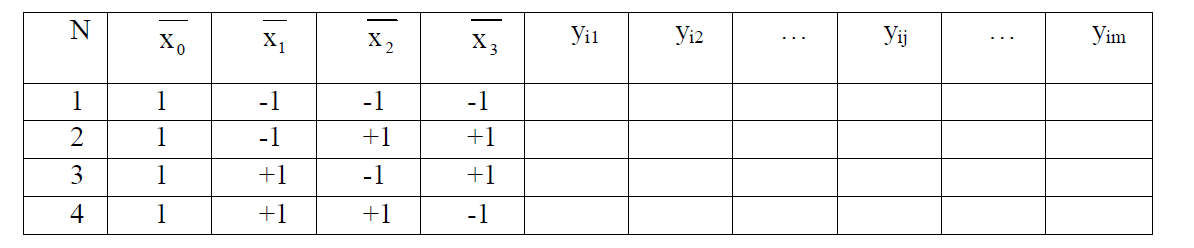


x1min = -40; x2min = -70; x3min = -20;

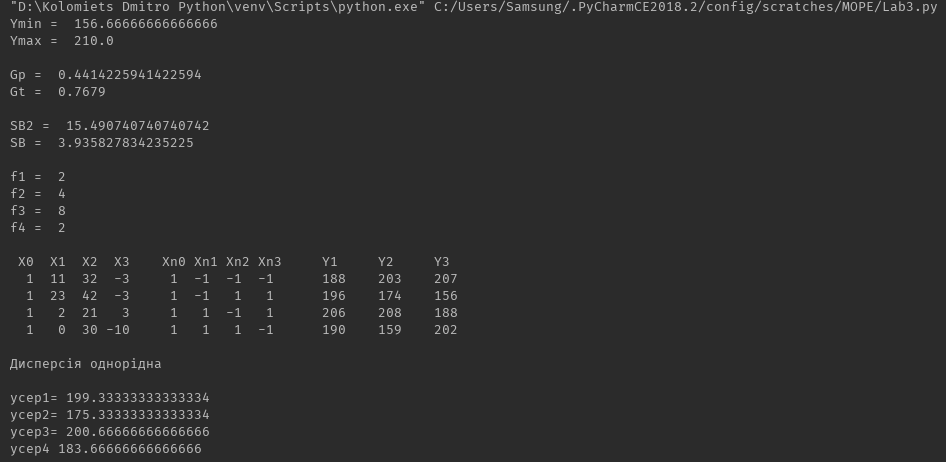
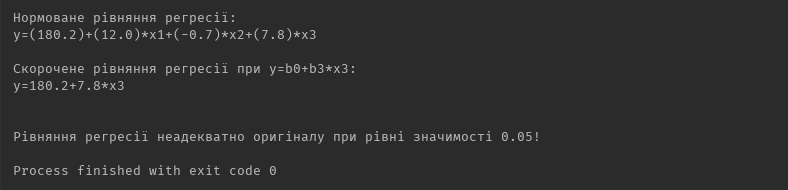
x1max = 20; x2min = -10; x3min = 20;

Ymin = 156.6 Ymax = 210

Нормована матриця планування:



1. Лістинг програми:
2. **import** numpy **as** np  
   **import** math  
     
   **"""  
   Вивід всіх значень я не зробив, але вивів остаточні значення.  
   Якщо потрібно буде все відформатувати, напишіть про це будь ласка !  
   Я закоментував що я робив, тому проблем не повинно виникати.  
   """**x1min = -40  
   x1max = 20  
     
   x2min = -70  
   x2max = -10  
     
   x3min = -20  
   x3max = 20  
     
   xsmin = (x1min + x2min + x3min)/3  
   xsmax = (x1max + x2max + x3max)/3  
     
   **def** column(matrix, i):  
    **return** [row[i] **for** row **in** matrix]  
     
   N = 4  
   m = 3  
   d = 2  
     
   X0s = np.ones(N, int)  
   X1s = np.random.randint(low=-10, high=50, size=N)  
   X2s = np.random.randint(low=20, high=60, size=N)  
   X3s = np.random.randint(low=-10, high=10, size=N)  
     
   ymax = 200 + xsmax  
   ymin = 200 + xsmin  
     
   Xs = [[0] \* 4 **for** i **in** range(N)]  
   **for** i **in** range(N):  
    Xs[i][0] = X0s[i]  
    Xs[i][1] = X1s[i]  
    Xs[i][2] = X2s[i]  
    Xs[i][3] = X3s[i]  
     
   Xsnorm = [[1, -1, -1, -1],  
    [1, -1, 1, 1],  
    [1, 1, -1, 1],  
    [1, 1, 1, -1]]  
     
   Yv = np.random.randint(low=ymin, high=ymax, size=m \* N).reshape(N, m)  
   *# Середнє значення функції відгуку в рядку*Ymid = []  
   **for** i **in** Yv:  
    Ymid.append(sum(i) / m)  
     
   *# Розрахунок нормованих коефіцієнтів рівняння регресії.*mx1 = sum(Xsnorm[1]) / N  
   mx2 = sum(Xsnorm[2]) / N  
   mx3 = sum(Xsnorm[3]) / N  
   my = sum(Ymid) / N  
     
   a1 = sum(np.multiply(Xsnorm[1], Ymid)) / N  
   a2 = sum(np.multiply(Xsnorm[2], Ymid)) / N  
   a3 = sum(np.multiply(Xsnorm[3], Ymid)) / N  
     
   a11 = sum(np.multiply(Xsnorm[1], Xsnorm[1])) / N  
   a22 = sum(np.multiply(Xsnorm[2], Xsnorm[2])) / N  
   a33 = sum(np.multiply(Xsnorm[3], Xsnorm[3])) / N  
     
   a12 = sum(np.multiply(Xsnorm[1], Xsnorm[2])) / N  
   a21 = a12  
   a13 = sum(np.multiply(Xsnorm[1], Xsnorm[3])) / N  
   a31 = a13  
   a23 = sum(np.multiply(Xsnorm[2], Xsnorm[3])) / N  
   a32 = a23  
     
   b0top = [[my, mx1, mx2, mx3], [a1, a11, a12, a13], [a2, a12, a22, a32], [a3, a13, a23, a33]]  
   b0bot = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]  
   b0 = np.linalg.det(b0top) / np.linalg.det(b0bot)  
     
   b1top = [[1, my, mx2, mx3], [mx1, a1, a12, a13], [mx2, a2, a22, a32], [mx3, a3, a23, a33]]  
   b1bot = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]  
   b1 = np.linalg.det(b1top) / np.linalg.det(b1bot)  
     
   b2top = [[1, mx1, my, mx3], [mx1, a11, a1, a13], [mx2, a12, a2, a32], [mx3, a13, a3, a33]]  
   b2bot = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]  
   b2 = np.linalg.det(b2top) / np.linalg.det(b2bot)  
     
   b3top = [[1, mx1, mx2, my], [mx1, a11, a12, a1], [mx2, a12, a22, a2], [mx3, a13, a23, a3]]  
   b3bot = [[1, mx1, mx2, mx3], [mx1, a11, a12, a13], [mx2, a12, a22, a32], [mx3, a13, a23, a33]]  
   b3 = np.linalg.det(b3top) / np.linalg.det(b3bot)  
     
   *# Дисперсія по рядках*desp = []  
   tick = 0  
   **for** i **in** Yv:  
    state = 0  
    **for** j **in** i:  
    state += (j - Ymid[tick]) \*\* 2  
    desp.append((state) / m)  
    tick += 1  
     
   **"""<================Критерій Кохрена==================>"""**Gp = max(desp) / sum(desp)  
   f1 = m - 1  
   f2 = N  
   Gt = 0.7679  
   print(**"Gp = "**,Gp,**"\nGt = "**,Gt)  
     
   **"""\<================Критерій Стьюдента==================>"""**Sb2 = sum(desp) / N  
   SB2 = Sb2 / (N \* m)  
   SB = math.sqrt(SB2)  
   print(**"\nSB2 = "**, SB2, **"\nSB = "**, SB )  
     
   beta0 = np.multiply(Ymid, Xsnorm[0])  
   Beta0 = sum(beta0) / N  
     
   beta1 = np.multiply(Ymid, Xsnorm[1])  
   Beta1 = sum(beta1) / N  
     
   beta2 = np.multiply(Ymid, Xsnorm[2])  
   Beta2 = sum(beta2) / N  
     
   beta3 = np.multiply(Ymid, Xsnorm[3])  
   Beta3 = sum(beta3) / N  
     
   t0 = math.fabs(Beta0) / SB  
   t1 = math.fabs(Beta1) / SB  
   t2 = math.fabs(Beta2) / SB  
   t3 = math.fabs(Beta3) / SB  
   t = 2.306  
     
   **if** t0>t:  
    trl=0  
   **else**: trl=1  
   **if** t1<t:  
    trl=0  
   **else**: trl=1  
   **if** t2<t:  
    trl=0  
   **else**: trl=1  
   **if** t3>t:  
    trl=0  
   **else**: trl=1  
     
   *# Виключаємо коефіцієнти b1, b2 з рівняння y = b0 + b3 \* x3***"""\<================Критерій Фішера==================>"""**f3 = f1 \* f2  
   f4 = N - d  
   print(**"\nf1 = "**,f1,**"\nf2 = "**, f2, **"\nf3 = "**, f3, **"\nf4 = "**, f4 )  
   yreg = []  
   **for** i **in** X3s:  
    yreg.append(b0 + b3 \* i)  
     
   sad = np.subtract(yreg, Ymid)  
   sad2 = np.multiply(sad, sad)  
   Sad2 = (m/(N-d))\*sum(sad2)  
     
   Fp = Sad2/Sb2  
   Ft = 4.5  
     
   *# Вивід даних*print(**"\n X0 X1 X2 X3 Xn0 Xn1 Xn2 Xn3 "**, end=**""**)  
   **for** i **in** range(m):  
    print(**"Y"**, i + 1, end=**" "**, sep=**""**)  
     
   print()  
     
   **for** i **in** range(N):  
    **for** j **in** range(len(Xs[i])):  
    print(**"{:3d}"**.format(Xs[i][j]), end=**" "**)  
    print(end=**" "**)  
    **for** j **in** range(len(Xsnorm[i])):  
    print(**"{:3d}"**.format(Xsnorm[i][j]), end=**" "**)  
    print(end=**" "**)  
    **for** j **in** range(m):  
    print(**"{:>6d}"**.format(Yv[i][j]), end=**" "**)  
    print()  
   print()  
     
   **if** Gp < Gt:  
    print(**"Дисперсія однорідна"**)  
   **else**:  
    print(**"Дисперсія не однорідна"**)  
    breakpoint()  
     
   print(**"\nусер1="**,Ymid[0] ,**"\nyсер2="**,Ymid[1],**"\nyсер3="**,Ymid[2],**"\nусер4"**,Ymid[3] )  
   print(**"\nНормоване рівняння регресії: \ny=({:.1f})+({:.1f})\*x1+({:.1f})\*x2+({:.1f})\*x3"**.format(b0,b1,b2,b3))  
   print(**"\nСкорочене рівняння регресії при у=b0+b3\*x3:"**)  
   print(**"y={:.1f}+{:.1f}\*x3"**.format(b0, b3))  
     
   **if** Fp>Ft:  
    print(**"\n\nРівняння регресії неадекватно оригіналу при рівні значимості 0.05!"**)  
   **else**: print(**"\n\nРівняння регресії адекватно оригіналу при рівні значимості 0.05!"**)
3. Результат виконання роботи програми:



**Контрольні запитання**

1. *Що називається дробовим факторним експериментом?*

У деяких випадках немає необхідності проводити повний факторний експеримент (ПФЕ). Якщо буде використовуватися лінійна регресія, то можливо зменшити кількість рядків матриці ПФЕ до кількості коефіцієнтів регресійної моделі. Кількість дослідів слід скоротити, використовуючи для планування так звані регулярні дробові репліки від повного факторного експерименту, що містять відповідну кількість дослідів і зберігають основні властивості матриці планування – це означає дробовий факторний експеримент (ДФЕ).

1. *Для чого потрібно розрахункове значення Кохрена?*

Для перевірки дисперсії на однорідність.

1. *Для чого перевіряється критерій Стьюдента?*

Для перевірки значущості коефіцієнтів регресії. Тобто, Якщо виконується нерівність *ts*< *tтабл*, то приймається нуль-гіпотеза, тобто вважається, що знайдений коефіцієнт *βs* є статистично незначущим і його слід виключити з рівняння регресії. Якщо *ts* > *tтабл* то гіпотеза не підтверджується, тобто *βs* – значимий коефіцієнт і він залишається в рівнянні регресії.

1. *Чим визначається критерій Фішера і як його застосовувати?*

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.

**Висновок:** Отже, у ході виконання лабораторної роботи № 3 провели дробовий трьохфакторний експеримент. Склали матрицю планування, знайшли

коефіцієнти рівняння регресії, провели 3 статистичні перевірки. Була написана текстова програма, результати наведені вище. Результати співпадають із калькулятором. Кінцева мета роботи досягнута!