

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

Методи планування експерименту Лабораторна робота №2
«ПРОВЕДЕННЯ ДВОФАКТОРНОГО ЕКСПЕРИМЕНТУ З
ВИКОРИСТАННЯМ ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

Виконав:
студент II курсу ФІОТ
групи ІВ-92
Скворцов П. С.
номер у списку групи – 21
Перевірив:
ас. Регіда П. Г.

Мета:

провести двофакторний експеримент, перевірити однорідність дисперсії за критерієм Романовського, отримати коефіцієнти рівняння регресії, провести натуралізацію рівняння регресії.

Завдання:

1. Записати лінійне рівняння регресії.
2. Обрати тип двофакторного експерименту і скласти матрицю планування для нього з використанням додаткового нульового фактору ($x_0=1$).
3. Провести експеримент в усіх точках повного факторного простору (знайти значення функції відгуку y). Значення функції відгуку задати випадковим чином у відповідності до варіанту у діапазоні $y_{\min} \div y_{\max}$
4. Перевірити однорідності дисперсії за критерієм Романовського
5. Знайти коефіцієнти нормованих рівнянь регресії і виконати перевірку (підставити значення нормованих факторів і коефіцієнтів у рівняння).
6. Провести натуралізацію рівняння регресії й виконати перевірку натуралізованого рівняння.
7. Написати комп'ютерну програму, яка все це виконує.

Варіант завдання:

№ _{варіанта}	X ₁		X ₂	
	min	max	min	max
221	10	40	10	60

Лістинг програми:

```
1 import math
2 from random import randint
3 import numpy as np
4
5 p_list = (0.99, 0.98, 0.95, 0.90)
6 rkr_table = {2: (1.73, 1.72, 1.71, 1.69),
7               6: (2.16, 2.13, 2.10, 2.00),
8               8: (2.43, 4.37, 2.27, 2.17),
9               10: (2.62, 2.54, 2.41, 2.29),
10              12: (2.75, 2.66, 2.52, 2.39),
11              15: (2.9, 2.8, 2.64, 2.49),
12              20: (3.08, 2.96, 2.78, 2.62)}
13
14 MIN_Y_LIMIT, MAX_Y_LIMIT, M = -10, 90, 5
15 X1_min, X1_max_n = 10, -1
16 X1_max, X1_max_n = 40, 1
17 X2_min, X2_min_n = 10, -1
18 X2_max, X2_max_n = 60, 1
19
20 Y_MATRIX = [[randint(MIN_Y_LIMIT, MAX_Y_LIMIT) for i in range(M)] for j in range(3)]
21
22 AVG_Y = [sum(Y_MATRIX[i][j] for j in range(M)) / M for i in range(3)]
23
24 SIGMA2_1 = sum([(j - AVG_Y[0]) ** 2 for j in Y_MATRIX[0]]) / M
25 SIGMA2_2 = sum([(j - AVG_Y[1]) ** 2 for j in Y_MATRIX[1]]) / M
26 SIGMA2_3 = sum([(j - AVG_Y[2]) ** 2 for j in Y_MATRIX[2]]) / M
27
28 SIGMA_TETA = math.sqrt((2 * (2 * M - 2)) / (M * (M - 4)))
29
30 FUV_1 = SIGMA2_1 / SIGMA2_2
31 FUV_2 = SIGMA2_3 / SIGMA2_1
32 FUV_3 = SIGMA2_3 / SIGMA2_2
33
34 TETA_UV_1 = ((M - 2) / M) * FUV_1
35 TETA_UV_2 = ((M - 2) / M) * FUV_2
36 TETA_UV_3 = ((M - 2) / M) * FUV_3
37
38 RUV1 = abs(TETA_UV_1 - 1) / SIGMA_TETA
39 RUV2 = abs(TETA_UV_2 - 1) / SIGMA_TETA
40 RUV3 = abs(TETA_UV_3 - 1) / SIGMA_TETA
41
42 MX1 = (-1 + 1 - 1) / 3
43 MX2 = (-1 - 1 + 1) / 3
44 MY = sum(AVG_Y) / 3
45 A1 = (1 + 1 + 1) / 3
46 A2 = (1 - 1 - 1) / 3
47 A3 = (1 + 1 + 1) / 3
48 A11 = (-1 * AVG_Y[0] + 1 * AVG_Y[1] - 1 * AVG_Y[2]) / 3
49 A22 = (-1 * AVG_Y[0] - 1 * AVG_Y[1] + 1 * AVG_Y[2]) / 3
50
51 B0 = np.linalg.det(np.dot([MY, MX1, MX2],
52                             [A11, A1, A2],
53                             [A22, A2, A3]),
54                     np.linalg.inv([1, MX1, MX2],
55                                     [MX1, A1, A2],
56                                     [MX2, A2, A3])))
57
58 B1 = np.linalg.det(np.dot([1, MY, MX2],
59                             [MX1, A11, A2],
60                             [MX2, A22, A3]),
61                     np.linalg.inv([1, MX1, MX2],
62                                     [MX1, A1, A2],
63                                     [MX2, A2, A3])))
64
65 B2 = np.linalg.det(np.dot([1, MX1, MY],
66                             [MX1, A1, A11],
67                             [MX2, A2, A22]),
68                     np.linalg.inv([1, MX1, MX2],
69                                     [MX1, A1, A2],
70                                     [MX2, A2, A3])))
71
72 def checkRegression():
73     NORM_Y_1 = round(B0 - B1 - B2, 1)
74     NORM_Y_2 = round(B0 + B1 - B2, 1)
```

```

75     NORM_Y_3 = round(B0 - B1 + B2, 1)
76     if NORM_Y_1 == AVG_Y[0] and NORM_Y_2 == AVG_Y[1] and NORM_Y_3 == AVG_Y[2]:
77         print("Значення перевірки нормованого рівняння регресії сходяться")
78     else:
79         print("Значення перевірки нормованого рівняння регресії НЕ сходяться")
80
81     NORM_Y = B0 - B1 + B2
82
83     DX1 = math.fabs(X1_max - X1_min) / 2
84     DX2 = math.fabs(X2_max - X2_min) / 2
85     X10 = (X1_max + X1_min) / 2
86     X20 = (X2_max + X2_min) / 2
87
88     AA0 = B0 - B1 * X10 / DX1 - B2 * X20 / DX2
89     AA1 = B1 / DX1
90     AA2 = B2 / DX2
91
92
93     def odnoridna_dispersion():
94         m = min(rkr_table, key=lambda x: abs(x - M))
95         p = 0
96         for ruv in (RUV1, RUV2, RUV3):
97             if ruv > rkr_table[m][0]:
98                 return False
99             for rkr in range(len(rkr_table[m])):
100                 if ruv < rkr_table[m][rkr]:
101                     p = rkr
102         return p_list[p]
103
104
105     def naturalized_regression(x1, x2):
106         return AA0 + AA1 * x1 + AA2 * x2
107
108
109     # output
110     for i in range(3):
111         print("Y{:}: {}, Average: {}".format(i + 1, Y_MATRIX[i], AVG_Y[i]))
112     print()
113     print("σ² y1:", SIGMA2_1)
114     print("σ² y2:", SIGMA2_2)
115     print("σ² y3:", SIGMA2_2)
116     print("σθ =", SIGMA_TETA)
117     print()
118     print("Fuv1 =", FUV_1)
119     print("Fuv2 =", FUV_2)
120     print("Fuv3 =", FUV_3)
121     print()
122     print("θuv1 =", TETA_UV_1)
123     print("θuv2 =", TETA_UV_2)
124     print("θuv3 =", TETA_UV_3)
125     print()
126     print("Ruv1 =", RUV1)
127     print("Ruv2 =", RUV2)
128     print("Ruv3 =", RUV3)
129     print()
130     print("Однорідна дисперсія:", odnoridna_dispersion())
131     print()
132     print("mx1:", MX1)
133     print("mx2:", MX2)
134     print("my:", MY)
135     print("a1:", A1)
136     print("a2:", A2)
137     print("a3:", A3)
138     print("a11:", A11)
139     print("a22:", A22)
140     print("b0:", B0)
141     print("b1:", B1)
142     print("b2:", B2)
143     print("Натуралізація коефіцієнтів:")
144     print("Δx1:", DX1)
145     print("Δx2:", DX2)
146     print("x10:", X10)
147     print("x20:", X20)
148     print("a0:", AA0)

```

```

149 print("a1:", AA1)
150 print("a2:", AA2)
151 print()
152 print("Натуралізоване рівняння регресії:")
153
154 NR_Y = [round(naturalized_regression(X1_min, X2_min), 2),
155         round(naturalized_regression(X1_max, X2_min), 2),
156         round(naturalized_regression(X1_min, X2_max), 2)]
157 print(NR_Y)
158 if NR_Y == AVG_Y:
159     print("Коефіцієнти натуралізованого рівняння регресії вірні")
160 else:
161     print("Коефіцієнти натуралізованого рівняння регресії НЕ вірні")
162 checkRegression()
163

```

Результати виконання роботи:

/Users/paul/.conda/envs/Skvortsov/bin/python /Users/paul/PycharmProjects/Skvortsov/MND/lab2.py

Y1: [80, 62, 52, 86, 82], Average: 72.4

Y2: [19, 33, 19, 55, 90], Average: 43.2

Y3: [88, 88, 28, -7, 38], Average: 47.0

$\sigma^2 y1$: 171.83999999999997

$\sigma^2 y2$: 720.96

$\sigma^2 y3$: 720.96

$\sigma\theta$ = 1.7888543819998317

Fuv1 = 0.2383488681757656

Fuv2 = 7.821229050279331

Fuv3 = 1.8641810918774966

θ_{uv1} = 0.14300932090545934

θ_{uv2} = 4.692737430167599

θ_{uv3} = 1.118508655126498

R_{uv1} = 0.4790723536347752

R_{uv2} = 2.064302979228158

R_{uv3} = 0.06624835219623211

Однорідна дисперсія: 0.9

mx1: -0.3333333333333333

mx2: -0.3333333333333333

my: 54.200000000000001

a1: 1.0

a2: -0.3333333333333333

a3: 1.0

a11: -25.400000000000002

a22: -22.866666666666667

b0: 45.10000000000000186

b1: -14.600000000000008

b2: -12.699999999999999

Натуралізація коефіцієнтів:

Δx_1 : 15.0

Δx_2 : 25.0

x_{10} : 25.0

x_{20} : 35.0

a0: 87.213333333333364

a1: -0.9733333333333386

a2: -0.5079999999999996

Натуралізоване рівняння регресії:

[72.4, 43.2, 47.0]

Коефіцієнти натуралізованого рівняння регресії вірні

Значення перевірки нормованого рівняння регресії сходяться

Process finished with exit code 0

Контрольні запитання:

1. Регресійні поліноми - це апроксимуючі поліноми, за допомогою яких ми можемо описати функцію. Застосовуються в теорії планування експерименту.
2. Однорідність дисперсії означає, що серед усіх дисперсій нема таких, які б значно перевищували одна одну.
3. Повний факторний експеримент - це такий факторний експеримент, коли використовуються усі можливі комбінації рівнів факторів.

Висновок:

У ході виконання лабораторної роботи було проведено двофакторний експеримент, перевірено однорідність дисперсії за критерієм Романовського, отримано коефіцієнти рівняння регресії та проведено натуралізацію рівняння регресії.