

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

Методи наукових досліджень

Лабораторна робота №2

«Проведення двофакторного есперименту з використанням
лінійного рівняння регресії»

Виконав:

студент 2 курсу, групи ІВ-91

Коренюк Андрій Олександрович

Залікова книжка № ІВ-9115

Варіант: 14

Перевірив: ас. Регіда П.Г.

Київ – 2021

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

Завдання

№ _{варіанта}	x_1		x_2	
	min	max	min	max
114	-25	75	25	65

Лістинг програми

```
from random import randint
from math import sqrt
from numpy.linalg import det
from beautifultable import BeautifulTable

"""Довірча ймовірність p = 0.9"""
romanovsky_table = {(2, 3, 4): 1.69, (5, 6, 7): 2.0, (8, 9): 2.17, (10, 11): 2.29,
                    (12, 13): 2.39, (14, 15, 16, 17): 2.49, (18, 19, 20): 2.62}

variant = {"n": 114, "x1min": -25, "x1max": 75, "x2min": 25, "x2max": 65}
y_min = (20 - variant["n"]) * 10
y_max = (30 - variant["n"]) * 10
x1 = [-1, -1, 1]
x2 = [-1, 1, -1]
nx1 = [variant["x1min"] if x1[i] == -1 else variant["x1max"] for i in range(3)]
nx2 = [variant["x2min"] if x2[i] == -1 else variant["x2max"] for i in range(3)]

m = 5
y_1 = [randint(y_min, y_max) for _ in range(m)]
y_2 = [randint(y_min, y_max) for _ in range(m)]
y_3 = [randint(y_min, y_max) for _ in range(m)]

y_average = [0, 0, 0]
y_dis = [0, 0, 0]
f_uv = [0, 0, 0]
sigma_uv = [0, 0, 0]
r_uv = [0, 0, 0]
main_divergence = 0
romanovsky_value = 0
```

```

def romanovsky_criterion():
    global y_average
    global y_dis
    global f_uv
    global sigma_uv
    global r_uv
    global main_divergence
    global romanovsky_value

    y_average = [sum(y_1)/m, sum(y_2)/m, sum(y_3)/m]
    y_dis = [0, 0, 0]

    for i in range(m):
        y_dis[0] += (y_1[i] - y_average[0]) ** 2
        y_dis[1] += (y_2[i] - y_average[1]) ** 2
        y_dis[2] += (y_3[i] - y_average[2]) ** 2

    for i in range(3):
        y_dis[i] /= m

    main_divergence = 2 * sqrt((m - 1)/(m * (m - 4)))
    uv_pairs = [[y_dis[0], y_dis[1]], [y_dis[1], y_dis[2]], [y_dis[2], y_dis[0]]]

    f_uv = list()
    sigma_uv = list()
    r_uv = list()
    for i in range(3):
        f_uv.append(max(uv_pairs[i]) / min(uv_pairs[i]))
        sigma_uv.append(f_uv[i] * (m - 2) / m)
        r_uv.append(abs(sigma_uv[i] - 1)/main_divergence)

    for key in romanovsky_table.keys():
        if m in key:
            romanovsky_value = romanovsky_table[key]
            break
    return max(r_uv) <= romanovsky_value

while not romanovsky_criterion():
    y_1.append(randint(y_min, y_max))
    y_2.append(randint(y_min, y_max))
    y_3.append(randint(y_min, y_max))
    m += 1

```

```
mx1, mx2, my = sum(x1) / 3, sum(x2) / 3, sum(y_average) / 3
a1, a2, a3, a11, a22 = 0, 0, 0, 0, 0
```

```
for i in range(3):
    a1 += x1[i] ** 2
    a2 += x1[i] * x2[i]
    a3 += x2[i] ** 2
    a11 += x1[i] * y_average[i]
    a22 += x2[i] * y_average[i]
```

```
a1, a2, a3, a11, a22 = a1 / 3, a2 / 3, a3 / 3, a11 / 3, a22 / 3
```

```
main_det = det([[1, mx1, mx2], [mx1, a1, a2], [mx2, a2, a3]])
B0 = det([[my, mx1, mx2], [a11, a1, a2], [a22, a2, a3]]) / main_det
B1 = det([[1, my, mx2], [mx1, a11, a2], [mx2, a22, a3]]) / main_det
B2 = det([[1, mx1, my], [mx1, a1, a11], [mx2, a2, a22]]) / main_det
```

```
dx1 = abs(variant["x1max"] - variant["x1min"]) / 2
dx2 = abs(variant["x2max"] - variant["x2min"]) / 2
x10 = (variant["x1max"] + variant["x1min"]) / 2
x20 = (variant["x2max"] + variant["x2min"]) / 2
```

```
A0 = B0 - B1 * x10 / dx1 - B2 * x20 / dx2
A1 = B1 / dx1
A2 = B2 / dx2
```

```
plan_m = BeautifulTable()
romanovsky_m = BeautifulTable()
rationed_checking_m = BeautifulTable()
natural_checking_m = BeautifulTable()
```

```
y_headers = [f"Y{i+1}" for i in range(m)]
plan_m.column_headers = ["№", "Rationed X1", "Rationed X2", *y_headers]
romanovsky_m.column_headers = ["№", "Average Y[j]", "Dispersion Y[j]", "Number uv",
"F[uv]", "θ[uv]", "R[uv]"]
rationed_checking_m.column_headers = ["№", "Rationed X1", "Rationed X2", "Average Y[j]", "Experimental Y[j]"]
natural_checking_m.column_headers = ["№", "Natured X1", "Natured X2", "Average Y[j]", "Experimental Y[j]"]
```

```
y = {0: y_1, 1: y_2, 2: y_3}
uv = [12, 23, 31]
for i in range(3):
    plan_m.append_row([i+1, x1[i], x2[i], *y[i]])
```

```

romanovsky_m.append_row([i+1, y_average[i], y_dis[i], uv[i], f_uv[i], sigma_uv[i], r_uv[i]])
rationed_checking_m.append_row([i+1, x1[i], x2[i], y_average[i], B0 + B1 * x1[i] + B2 *
x2[i]])
natural_checking_m.append_row([i+1, nx1[i], nx2[i], y_average[i], A0 + A1 * nx1[i] + A2 *
nx2[i]])

print(plan_m, "\n")
print(romanovsky_m)
print(f"Main Divergence: {main_divergence}")
print(f"Romanovsky Criterion: {romanovsky_value}", "\n")
print(f"y = {B0} + {B1} * x1 + {B2} * x2")
print(rationed_checking_m, "\n")
print(f"y = {A0} + {A1} * x1 + {A2} * x2")
print(natural_checking_m)

```

Результат виконання роботи

№	Rationed X1	Rationed X2	Y1	Y2	Y3	Y4	Y5
1	-1	-1	-849	-850	-878	-937	-885
2	-1	1	-926	-883	-924	-891	-858
3	1	-1	-915	-860	-932	-855	-896

№	Average Y[j]	Dispersion Y[j]	Number uv	F[uv]	θ[uv]	R[uv]
1	-879.8	1027.76	12	1.547	0.928	0.04
2	-896.4	664.24	23	1.366	0.82	0.101
3	-891.6	907.44	31	1.133	0.68	0.179

Main Divergence: 1.7888543819998317

Romanovsky Criterion: 2.0

$$y = -893.9999999999998 + -5.899999999999919 * x_1 + -8.299999999999987 * x_2$$

№	Rationed X1	Rationed X2	Average Y[j]	Experimental Y[j]
1	-1	-1	-879.8	-879.8
2	-1	1	-896.4	-896.4
3	1	-1	-891.6	-891.6

$$y = -872.3750000000001 + -0.11799999999999837 * x_1 + -0.4149999999999935 * x_2$$

№	Natured X1	Natured X2	Average Y[j]	Experimental Y[j]
1	-25	25	-879.8	-879.8
2	-25	65	-896.4	-896.4
3	75	25	-891.6	-891.6

Відповіді на контрольні запитання

1. Що таке регресійні поліноми і де вони застосовуються?

Відповідь: нехай існує теоретичний ряд, що описує залежність функції відгуку від k факторів $\eta = \varphi(x_1, x_2, \dots, x_k)$. Його коефіцієнти розрахувати неможливо.

Регресійний поліном – це поліном, що замінює даний теоретичний ряд та має коефіцієнти, які можна розрахувати експериментально. Регресійний поліном застосовується при проведенні будь-якого експерименту.

2. Визначення однорідності дисперсії.

Відповідь: однорідність дисперсії – це необхідна умова підтвердження гіпотези про забезпечення нормального закону розподілу вимірюваної величини при вибраній кількості повторів m та вибраній ймовірності p .

Якщо $N = 2$ ($k = 1$), то для перевірки однорідності дисперсії використовується критерій Фішера або критерій Романовського. Якщо $N \geq 3$ ($k \geq 2$), то для перевірки однорідності дисперсії використовується критерій Кохрена.

3. Що називається повним факторним експериментом?

Відповідь: це експеримент, у якому використовуються всі можливі комбінації рівнів факторів (r^k).