Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Теоретической и прикладной математики

Лабораторная работа № 1,2

по дисциплине «Статистические методы анализа данных»

Факультет:	ПМИ
Группа:	ПМИ-02
Вариант:	6
Студент:	Сидоров Даниил,
	Дюков Богдан
Преподаватель:	Попов Александр Александрович.

Новосибирск

2026

1. Постановка задачи

Провести моделирование объекта, о котором известно: число действующих факторов – три; по всем факторам зависимость выхода близка к линейной, взаимодействия первого фактора со вторым и третьим существенны, т. е. соответствующие параметры θ при регрессорах x_1x_2, x_1x_3 значительно отличаются от нулевого значения. Также известно, что первый фактор в эксперименте может варьироваться на трех уровнях, второй фактор варьируется на четырех уровнях, третий фактор на двух уровнях.

Спроектировать и сформировать программные модули по вычислению МНК-оценок параметров для заданной параметрической модели объекта. Предусмотреть достаточно простой способ настройки программы на необходимый вид (структуру) модели. Пользуясь экспериментальными данными, полученными в лабораторной работе № 1, оценить параметры модели объекта. Проверить адекватность полученной модели

2. Описание объекта

Выберем линейную относительно параметров θ имитационную модель $\eta(x, \theta)$:

$$u = \eta(x, \theta) = \theta^T f(x_1, x_2, x_3) =$$

$$= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1 x_2 + \theta_5 x_1 x_3 + \theta_6 x_1^2 + \theta_7 x_2^2 + \theta_8 x_3^2;$$

Определим параметры модели. Так как по всем факторам зависимость выхода близка к линейной, то значения $\theta_1, \theta_2, \theta_3$ должны быть достаточно велики. Также по условию задачи взаимодействия первого фактора со вторым и третьим существенны, поэтому значения θ_5 и θ_6 тоже достаточно большие. Отсюда имеем следующие значения параметров θ:

$$\theta = (1, 2, 5, 4, 1.5, 2.5, 0.02, 0.01, 0.03)^T;$$

Зададим области определения для трех факторов в соответствии с заданными уровнями (при условии, что $x_i \in [-1, +1], i = 1, \dots, k$):

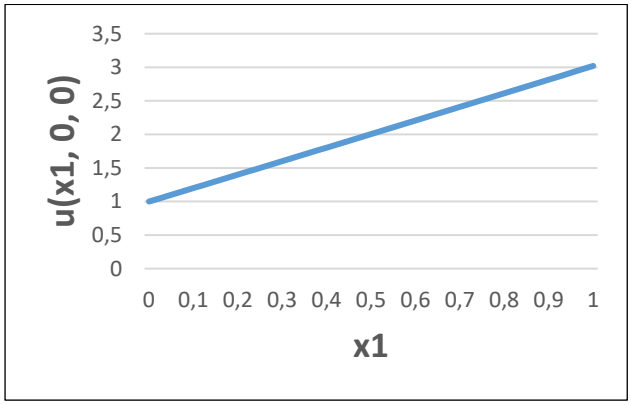
$$x_1 = \{-1, 0, 1\};$$

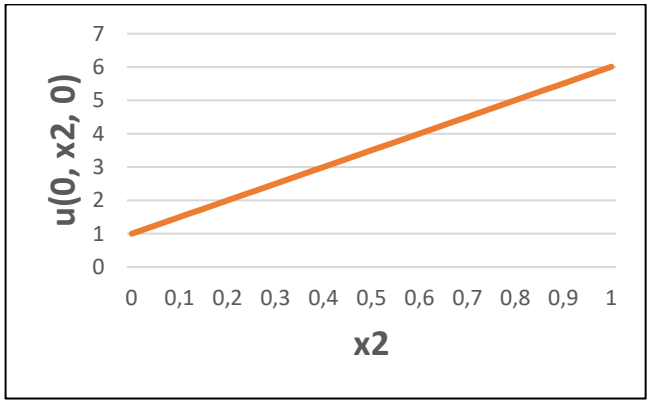
$$x_2 = \{-1, -0.33, 0.33, 1\};$$

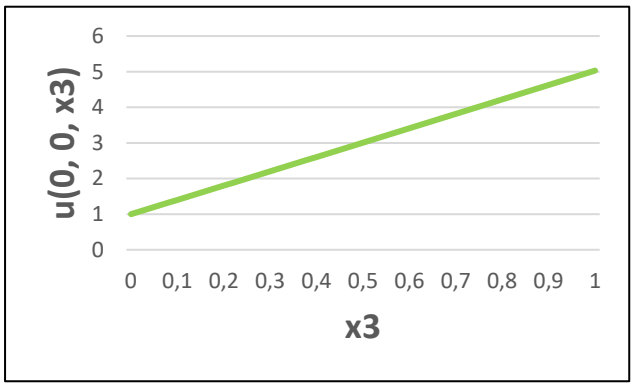
$$x_3 = \{-0.9, 0.9\}.$$

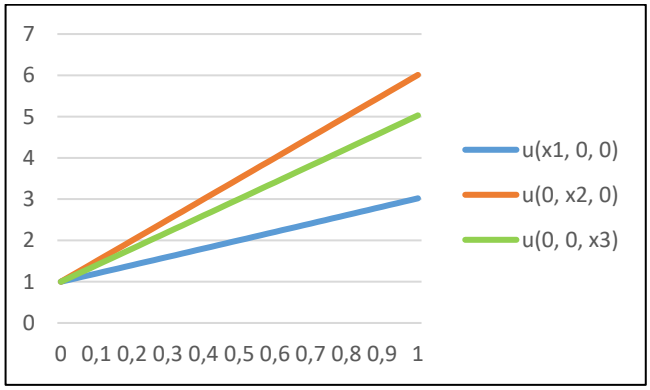
Выберем необходимое число экспериментов n. Оно должно быть как минимум в 2-3 раза превышать число оцениваемых параметров модели (число параметров модели – это размерность вектора θ). В нашем случае n = 24.

3. Графики зависимости незашумленного отклика от входных факторов









4. Генерация данных

Значение отклика y_i будем считать по формуле:

$$y_i = u_i + e_i = \eta(x_i, \theta) + e_i;$$

Генерацию значения ошибок наблюдений e_i будем делать по нормальному закону с нулевым математическим ожиданием и дисперсией равной доли $\rho = 5\%$ от мощности сигнала w^2:

$$n = 24;$$

$$\rho = 0.05;$$

$$w^2 = \frac{(u-\bar{u})^T(u-\bar{u})}{n-1} = 33.77, \text{где } \bar{u} = \frac{1}{n}\sum_{j=1}^{n} u_j = 0.22;$$

$$\sigma^2 = \rho w^2 = 1.69;$$

$$e_i \sim N(0, \sigma^2);$$

В результате имеем следующую таблицу:

№	x1	x2	x3	u	e	y
1	-1	-1	-0,9	-5,7957	2,044147661	-3,751552339
2	-1	-1	0,9	-3,0957	-3,718553868	-6,814253868
3	-1	-0,33	-0,9	-3,459611	0,062949987	-3,396661013
4	-1	-0,33	0,9	-0,759611	1,144543296	0,384932296
5	-1	-0,33	-0,9	-3,459611	0,487447636	-2,972163364
6	-1	-0,33	0,9	-0,759611	0,09538094	-0,66423006
7	-1	1	-0,9	1,2043	0,365898648	1,570198648
8	-1	1	0,9	3,9043	-0,656855041	3,247444959
9	0	-1	-0,9	-7,5657	0,918960825	-6,646739175
10	0	-1	0,9	-0,3657	-0,255801593	-0,621501593
11	0	-0,33	-0,9	-4,224611	-0,687023272	-4,911634272
12	0	-0,33	0,9	2,975389	1,251657066	4,227046066
13	0	-0,33	-0,9	-4,224611	-1,098553144	-5,323164144
14	0	-0,33	0,9	2,975389	-0,642437777	2,332951223
15	0	1	-0,9	2,4343	-0,711016794	1,723283206
16	0	1	0,9	9,6343	0,03367812	9,66797812
17	1	-1	-0,9	-9,2957	-1,519921104	-10,8156211
18	1	-1	0,9	2,4043	-0,836669198	1,567630802
19	1	-0,33	-0,9	-4,949611	1,35949113	-3,59011987
20	1	-0,33	0,9	6,750389	0,731955191	7,482344191
21	1	-0,33	-0,9	-4,949611	-0,757562701	-5,707173701
22	1	-0,33	0,9	6,750389	-0,692406518	6,057982482
23	1	1	-0,9	3,7043	-0,885124176	2,819175824
24	1	1	0,9	15,4043	1,507168274	16,91146827

5. Оценка параметров

Оценка параметров выполняется с помощью метода наименьших квадратов:

$$\hat{\theta} = \arg\min\left(y - \theta^{\mathrm{T}} f(x)\right)^{\mathrm{T}}\left(y - y - \theta^{\mathrm{T}} f(x)\right) = (X^T X)^{-1} X y;$$

Несмещенная оценка $\hat{\sigma}^2$ неизвестной дисперсии наблюдения равна:

$$\hat{\sigma}^2 = \frac{\hat{e}^T \hat{e}}{n - m}, \text{где вектор остатков } \hat{e} = y - \hat{y} = y - X\hat{\theta};$$

Полученные оценки:

$$\hat{\theta} = (1.48, 1.98, 5.25, 3.93, 1.72, 3.1, 0.09, -0.55, -0.7)^T;$$

$$\hat{\sigma}^2 = 1.99;$$

Оценка $\widehat{\sigma}^2$ позволяет выполнить проверку гипотезы об адекватности модели. При $\alpha = 0.05$, большей дисперсии $f = \infty$, меньшей дисперсии $n - m = 24 - 8 = 16$ значение квантили $F_T = 1.67$. Вычисление F -статистики:

$$F = \frac{\widehat{\sigma}^2}{\sigma^2} = 1.18 < F_T;$$

Модель является адекватной.

y	y_hat	y-y_hat	theta	theta_hat	sigma_squared	sigma_hat_squared
-3,751552339	-5,802964946	2,051412607	1	1,477979448	1,688422277	1,986457977
-6,814253868	-4,313937914	-2,500315955	2	1,978277684		
-3,396661013	-2,945728282	-0,450932731	5	5,251798859		
0,384932296	-1,456701249	1,841633546	4	3,925090935		
-2,972163364	-2,945728282	-0,026435082	1,5	1,716087614		
-0,66423006	-1,456701249	0,79247119	2,5	3,097853695		
1,570198648	1,268457544	0,301741104	0,02	0,089560206		
3,247444959	2,757484577	0,489960383	0,01	-0,547985782		
-6,646739175	-8,418403407	1,771664233	0,03	-0,69631651		
-0,621501593	-1,353239724	0,731738132				
-4,911634272	-4,411388042	-0,500246231				
4,227046066	2,653775641	1,573270425				
-5,323164144	-4,411388042	-0,911776102				
2,332951223	2,653775641	-0,320824418				
1,723283206	2,08519431	-0,361911104				
9,66797812	9,150357993	0,517620126				
-10,8156211	-10,85472146	0,039100353				
1,567630802	1,786578877	-0,218948074				
-3,59011987	-5,69792739	2,10780752				
7,482344191	6,943372943	0,538971247				
-5,707173701	-5,69792739	-0,009246311				
6,057982482	6,943372943	-0,885390461				
2,819175824	3,081051489	-0,261875665				
16,91146827	15,72235182	1,189116452				

6. Код программы

```
import numpy as np

import pandas as pd


# Генерация комбинаций факторов

def generate_combinations(x1_levels, x2_levels, x3_levels):

    x1_list = []

    x2_list = []

    x3_list = []
```

```
    for i in x1_levels:
        for j in x2_levels:
            for k in x3_levels:
                x1_list.append(i)
                x2_list.append(j)
                x3_list.append(k)

    return map(np.array, [x1_list, x2_list, x3_list])

# Сохранение датафрейма df в файл csv с именем filename
def save_to_csv(df, filename):
    df.to_csv(filename, index=False)

# Получение датафрейма с данными, необходимыми для построения графиков
def get_dataframe_for_graphs(theta):
    # Создание нового диапазона значений для каждого фактора с заданным шагом
    step = 0.02
    x1_range = np.arange(-1, 1 + step, step)
    x2_range = np.arange(-1, 1 + step, step)
    x3_range = np.arange(-1, 1 + step, step)

    # Вычисление отклика без шума для каждого фактора при нулевых значениях остальных факторов
    u_x1_0_0_range = theta[0] + theta[1] * x1_range + theta[6] * x1_range ** 2
    u_0_x2_0_range = theta[0] + theta[2] * x2_range + theta[7] * x2_range ** 2
    u_0_0_x3_range = theta[0] + theta[3] * x3_range + theta[8] * x3_range ** 2

    return pd.DataFrame({
        'x1': x1_range,
        'x2': x2_range,
        'x3': x3_range,
        'u(x1, 0, 0)': u_x1_0_0_range,
        'u(0, x2, 0)': u_0_x2_0_range,
        'u(0, 0, x3)': u_0_0_x3_range
    })

# Получение дисперсии шума
def get_sigma_squared(u):
    # Вычисление мощности сигнала
    omega_squared = np.dot(u - np.mean(u), u - np.mean(u)) / (len(u) - 1)
    # Доля от мощности сигнала
    rho = 0.05
```

```
    # Вычисление дисперсии шума
    return rho * omega_squared

# Получение ошибки
def get_noise(u, sigma_squared):
    return np.random.normal(0, np.sqrt(sigma_squared), len(u))

# Определение параметров
theta = np.array([1, 2, 5, 4, 1.5, 2.5, 0.02, 0.01, 0.03])

# Определение уровней для каждого фактора
x1_levels = np.array([-1, 0, 1])
x2_levels = np.array([-1, -0.33, -0.33, 1])
x3_levels = np.array([-0.9, 0.9])

# Генерация комбинаций факторов
x1, x2, x3 = generate_combinations(x1_levels, x2_levels, x3_levels)

# Вычисление истинного отклика без шума
u = theta[0] + theta[1]*x1 + theta[2]*x2 + theta[3]*x3 \
    + theta[4]*x1*x2 + theta[5]*x1*x3 \
    + theta[6]*x1**2 + theta[7]*x2**2 + theta[8]*x3**2

# Вычисление дисперсии шума
sigma_squared = get_sigma_squared(u)

# Вычисление ошибки
e = get_noise(u, sigma_squared)

# Зашумленный отклик
y = u + e

# Создание матрицы X
X = np.column_stack((np.ones(len(x1)), x1, x2, x3, x1*x2, x1*x3, x1**2, x2**2, x3**2))

# Вычисление вектора параметров модели theta_hat
theta_hat = np.linalg.inv(X.T @ X) @ X.T @ y

y_hat = X @ theta_hat

# Вычисление вектора остатков
e_hat = y - y_hat
```

```
# Вычисление несмещенной оценки
sigma_hat_squared = e_hat.T @ e_hat / (len(y) - len(theta))

# Вычисление F-статистики
F=sigma_hat_squared/sigma_squared

# Сохранение всех датафреймов в файлы csv
save_to_csv(pd.DataFrame({'x1': x1, 'x2': x2, 'x3': x3, 'u': u, 'e': e, 'y': y}), 'ResultsForLab1.csv')
save_to_csv(get_dataframe_for_graphs(theta), 'DataForGraphs.csv')
save_to_csv(pd.concat([
    pd.DataFrame({'y': y}),
    pd.DataFrame({'y_hat': y_hat}),
    pd.DataFrame({'y-y_hat': y-y_hat}),
    pd.DataFrame({'theta': theta}),
    pd.DataFrame({'theta_hat': theta_hat}),
    pd.DataFrame({'sigma_squared': [sigma_squared]}),
    pd.DataFrame({'sigma_hat_squared': [sigma_hat_squared]}),
    pd.DataFrame({'F': [F]})], axis=1), 'ResultsForLab2.csv')clude <stdio.h>
```