ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, к. т. н., доцент |  |  |  | В. В. Мышко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 |
| Многофакторный регрессионный анализ |
| по курсу: Обработка экспериментальных данных |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4236 |  |  |  | Л. Мвале |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2026

1. **Цель**

На основе заданного массива данных построить уравнение регрессии в виде линейного алгебраического полинома от двух переменных, проверить адекватность уравнения регрессии и значимость факторов регрессии. Расчеты произвести в матричной форме.

1. **Задание**

На основе заданного массива данных:

1. построить уравнение регрессии в виде *линейного алгебраического полинома от двух переменных;*
2. проверить адекватность уравнения регрессии;
3. проверить значимость факторов регрессии.

Расчеты произвести в матричной форме.

**Порядок выполнения задания:**

1. Выполнить центрирование факторов (массив экспериментальных данных, таблица 5.1);
2. Составить матричное уравнение с вектором неизвестных оценок коэффициентов регрессии;
3. Найти оценки коэффициентов регрессии посредством решения матричного уравнения;
4. Проверить адекватность построенного уравнения регрессии экспериментальным данным по критерию Фишера при уровне значимости α = 0,05;
5. Выполнить селекцию факторов по критерию Стьюдента при таком же уровне значимости;
6. Повторно проверить адекватность уравнения регрессии после исключения незначимых факторов.

Таблица 1

**Экспериментальные данные**

| № | № в группе | Массив экспериментальных данных | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 97 | 4236-97 | x1 | 0,5 | 1 | 2 | 4 | 3,5 | -1 |
| x2 | 0 | 1 | 1,5 | 2 | 2,5 | 1 |
| y | 0,2 | 5 | 8,8 | 15,2 | 17 | 1 |

Объем выборки: n = 6

Уравнение регрессии: линейный полином от двух переменных

wps

Уровень значимости: α = 0,05

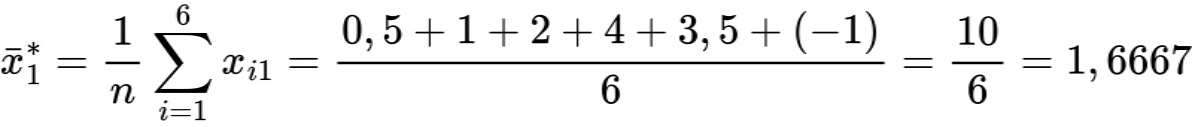
**Решение**

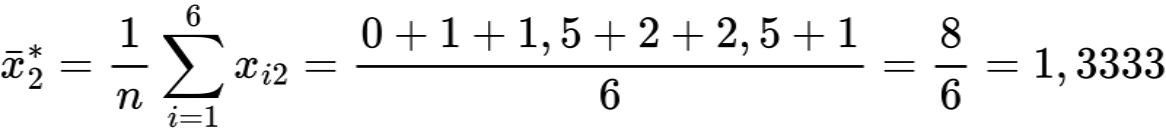
1. Центрирование факторов

Согласно §9.3.2, для упрощения вычислений используем центрированные значения факторов по формуле (9.3.5):

wps

Вычисляем средние значения факторов:



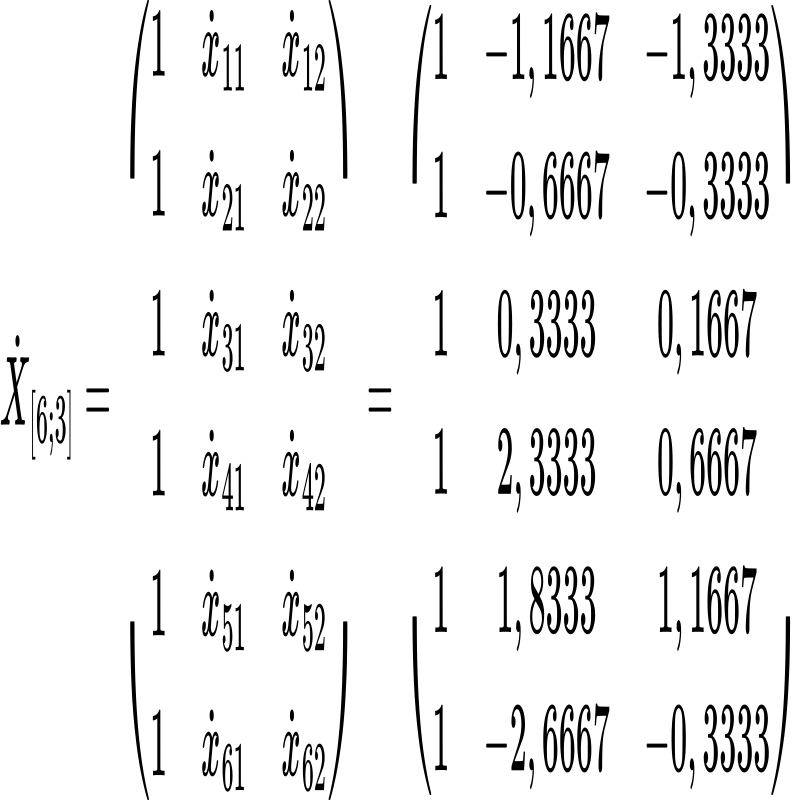


Выполняем центрирование факторов:

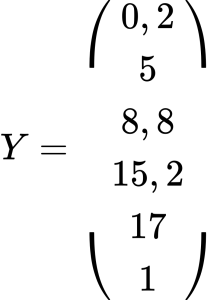
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i** | **x₁** | **x₂** | **y** | **wps** | **wps** |
| 1 | 0,5 | 0 | 0,2 | 0,5 - 1,6667 = -1,1667 | 0 - 1,3333 = -1,3333 |
| 2 | 1 | 1 | 5 | 1 - 1,6667 = -0,6667 | 1 - 1,3333 = -0,3333 |
| 3 | 2 | 1,5 | 8,8 | 2 - 1,6667 = 0,3333 | 1,5 - 1,3333 = 0,1667 |
| 4 | 4 | 2 | 15,2 | 4 - 1,6667 = 2,3333 | 2 - 1,3333 = 0,6667 |
| 5 | 3,5 | 2,5 | 17 | 3,5 - 1,6667 = 1,8333 | 2,5 - 1,3333 = 1,1667 |
| 6 | -1 | 1 | 1 | TRUE | 1 - 1,3333 = -0,3333 |

2. Составление матричного уравнения

Для линейной модели множественной регрессии с центрированными факторами матрица wpsимеет вид (согласно формуле 9.3.6):



Вектор наблюдений Y:

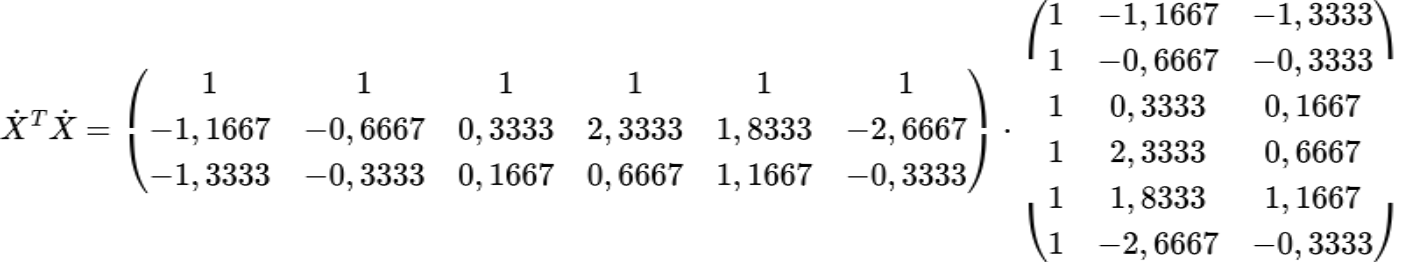


Матричное уравнение для нахождения оценок коэффициентов регрессии (согласно формуле 9.3.15):

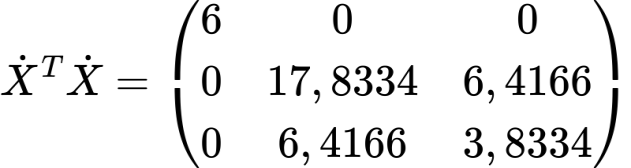
wps

где wps — вектор оценок коэффициентов регрессии.

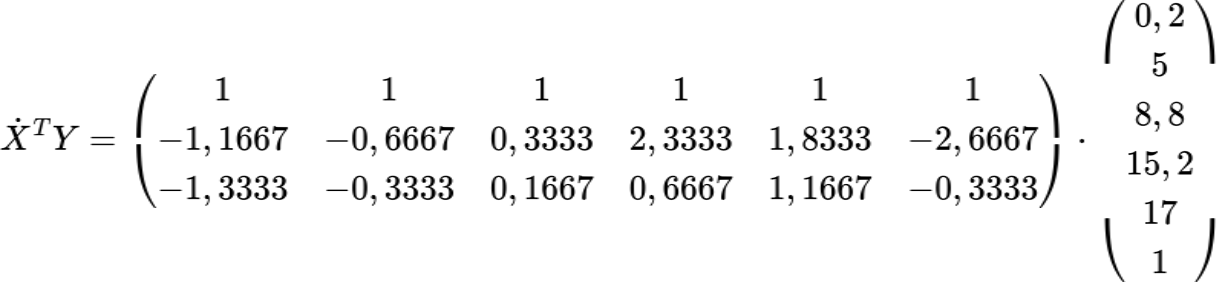
3. Вычисление матрицы wps



После вычисление



4. Вычисление вектора wps



Вычисляем компоненты:

Первая компонента:

wps

Вторая компонента:

wps

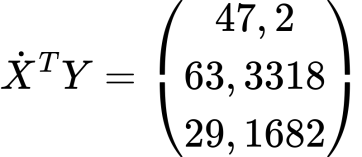
wps

Третья компонента:

wps

wps

Таким образом:



5. Решение матричного уравнения

Решение матричного уравнения имеет вид (согласно формуле 9.3.16):

wps

5.1. Вычисление определителя матрицы wps

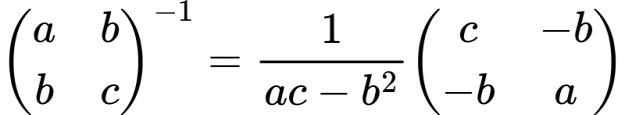


wps

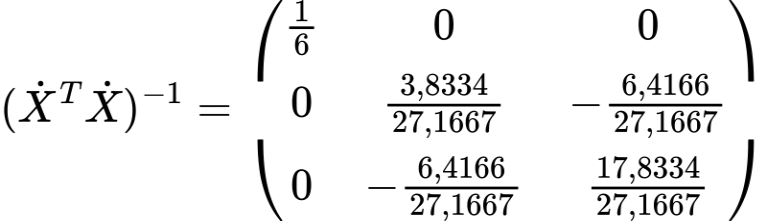
wps

5.2. Вычисление обратной матрицы

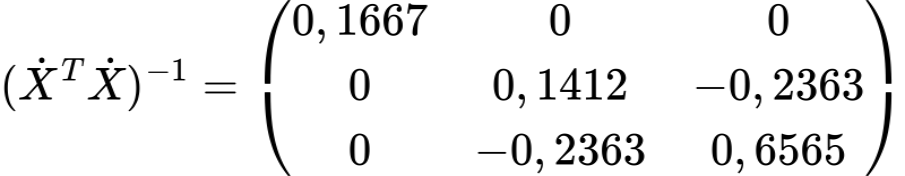
Для матрицы вида wps обратная матрица вычисляется по формуле:



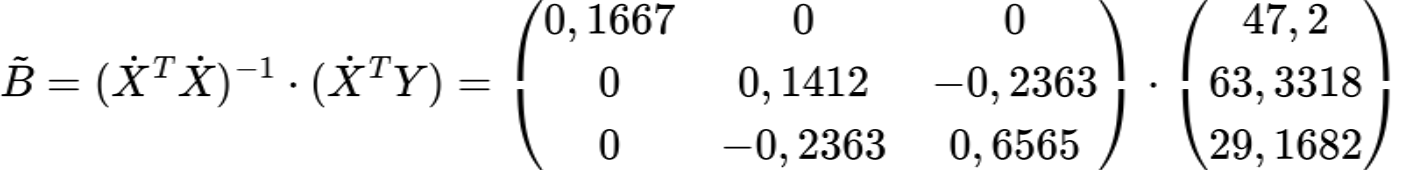
Таким образом:



Таким образом:



5.3. Вычисление оценок коэффициентов регрессии



Для wps:

wps

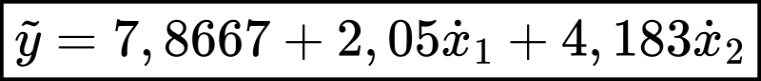
Для C:/Users/1/AppData/Local/Temp/wps.VHREdOwps:

wps

Для C:/Users/1/AppData/Local/Temp/wps.yFFsVLwps:

wps

Таким образом, уравнение регрессии в центрированных переменных:



5.4. Переход к исходным переменным

Для перехода к исходным переменным используем соотношение (9.3.5):

wps

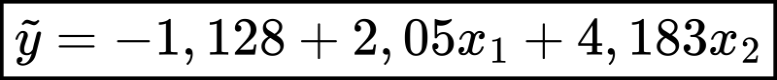
Подставляем:

wps

wps

wps

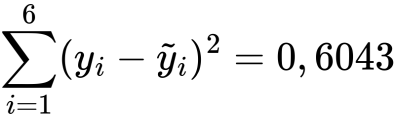
Уравнение регрессии в исходных переменных:



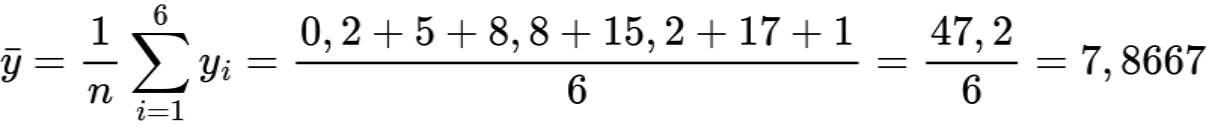
6. Проверка адекватности уравнения регрессии по критерию Фишера

6.1. Вычисление значений wpsДля каждого наблюдения вычисляем wpsпо полученному уравнению:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **i** | **x₁** | **x₂** | **y** | **wps** | **wps** | **wps** |
| 1 | 0,5 | 0 | 0,2 | -0,103 | 0,2 - (-0,103) = 0,303 | 0,0918 |
| 2 | 1 | 1 | 5 | 5,105 | 5 - 5,105 = -0,105 | 0,011 |
| 3 | 2 | 1,5 | 8,8 | 9,2465 | 8,8 - 9,2465 = -0,4465 | 0,1994 |
| 4 | 4 | 2 | 15,2 | 15,438 | 15,2 - 15,438 = -0,238 | 0,0566 |
| 5 | 3,5 | 2,5 | 17 | 16,5045 | 17 - 16,5045 = 0,4955 | 0,2455 |
| 6 | -1 | 1 | 1 | 1,005 | 1 - 1,005 = -0,005 | 0,000025 |
| Σ |  |  |  |  |  | 0,6043 |

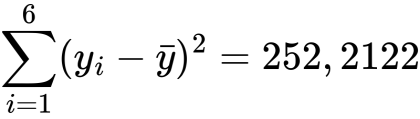


6.2. Вычисление среднего значения wps



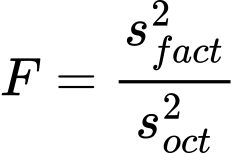
6.3. Вычисление сумм квадратов

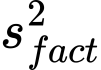
|  |  |  |  |
| --- | --- | --- | --- |
| **i** | **yᵢ** | **wps** | **wps** |
| 1 | 0,2 | 0,2 - 7,8667 = -7,6667 | 58,7778 |
| 2 | 5 | 5 - 7,8667 = -2,8667 | 8,218 |
| 3 | 8,8 | 8,8 - 7,8667 = 0,9333 | 0,8711 |
| 4 | 15,2 | 15,2 - 7,8667 = 7,3333 | 53,7778 |
| 5 | 17 | 17 - 7,8667 = 9,1333 | 83,4164 |
| 6 | 1 | 1 - 7,8667 = -6,8667 | 47,1511 |
| Σ |  |  | 252,2122 |



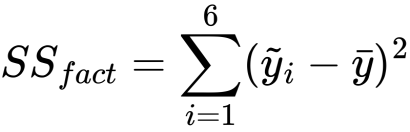
6.4. Вычисление факторной и остаточной дисперсий

Согласно §9.3.3, для проверки адекватности уравнения регрессии используется критерий Фишера:



где  — факторная дисперсия (дисперсия, обусловленная регрессией), wps — остаточная дисперсия.

Вычисляем факторную сумму квадратов:

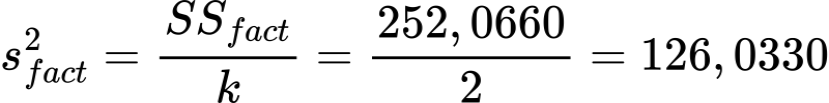


|  |  |  |  |
| --- | --- | --- | --- |
| i | wps | wps | wps |
| 1 | -0,103 | TRUE | 63,5161 |
| 2 | 5,105 | 5,105 - 7,8667 = -2,7617 | 7,627 |
| 3 | 9,2465 | 9,2465 - 7,8667 = 1,3798 | 1,9038 |
| 4 | 15,438 | 15,438 - 7,8667 = 7,5713 | 57,3246 |
| 5 | 16,5045 | 16,5045 - 7,8667 = 8,6378 | 74,6116 |
| 6 | 1,005 | 1,005 - 7,8667 = -6,8617 | 47,0829 |
| Σ |  |  | 252,066 |

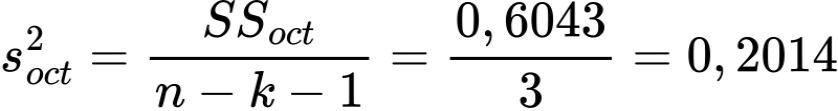


Проверка: wps (небольшое расхождение связано с округлениями).

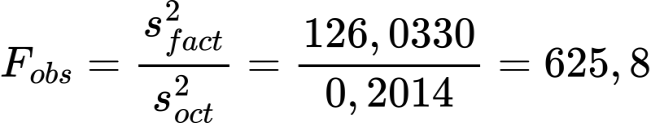
Факторная дисперсия (с числом степеней свободы k = 2):



Остаточная дисперсия (с числом степеней свободы n - k - 1 = 3):



6.5. Вычисление наблюдаемого значения критерия Фишера



6.6. Определение критического значения критерия Фишера

При уровне значимости wps и степенях свободы:

wps

wps

По таблице критических точек распределения Фишера (Приложение 5) находим:

wps

6.7. Сравнение и вывод

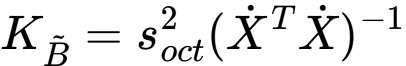
wps

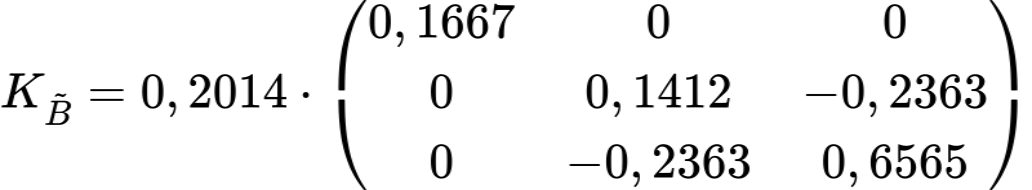
Так как wps, нулевая гипотеза об адекватности уравнения регрессии принимается. Уравнение регрессии адекватно описывает экспериментальные данные.

7. Проверка значимости факторов по критерию Стьюдента

7.1. Вычисление корреляционной матрицы коэффициентов регрессии

Согласно формуле (9.3.22), корреляционная матрица вектора оценок коэффициентов регрессии:





Вычисляем дисперсии коэффициентов (элементы на главной диагонали):

wps

wps

wps

7.2. Вычисление средних квадратических отклонений коэффициентов

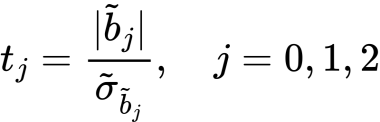
wps

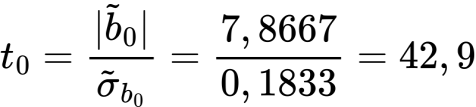
wps

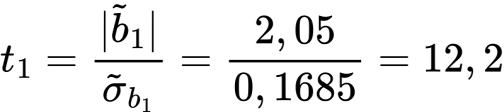
wps

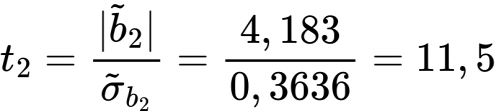
7.3. Вычисление наблюдаемых значений критерия Стьюдента

Согласно формуле (9.3.21)









7.4. Определение критического значения критерия Стьюдента

При уровне значимости wps и числе степеней свободыwps по таблице критических точек распределения Стьюдента (Приложение 6) находим:

wps

7.5. Сравнение и вывод

Согласно условию (9.3.23), коэффициент значим, если wps:

wps

wps

wps

Все наблюдаемые значения критерия Стьюдента больше критического значения. Это означает, что все коэффициенты регрессии являются значимыми на уровне значимости wps. Следовательно, оба фактора wps и C:/Users/1/AppData/Local/Temp/wps.thBgwqwps оказывают существенное влияние на зависимую переменную y.

8. Повторная проверка адекватности после исключения незначимых факторов

Так как все коэффициенты оказались значимыми, исключение факторов не требуется. Уравнение регрессии остается без изменений:

wps

Проверка адекватности уже выполнена в п.6 и подтвердила адекватность модели.

9. Итоговые результаты

|  |  |
| --- | --- |
| **Показатель** | **Значение** |
| Уравнение регрессии | wps |
| Общая сумма квадратов wps | 252,2122 |
| Факторная сумма квадратов wps | 252,066 |
| Остаточная сумма квадратов wps | 0,6043 |
| Факторная дисперсия wps | 126,033 |
| Остаточная дисперсия wps | 0,2014 |
| Наблюдаемое значение wps | 625,8 |
| Критическое значениеwps | 9,55 |
| Результат проверки адекватности | Модель адекватна |
| wps для wps | 42,9 |
| C:/Users/1/AppData/Local/Temp/wps.ZVbsAJwps для C:/Users/1/AppData/Local/Temp/wps.uTwJubwps | 12,2 |
| C:/Users/1/AppData/Local/Temp/wps.NfsfOcwps для C:/Users/1/AppData/Local/Temp/wps.TDdQOSwps | 11,5 |
| Критическое значение wps | 3,182 |
| Результат проверки значимости | Все коэффициенты значимы |

**Результаты работы**

В ходе выполнения данной лабораторной работы была написана программа на языке Python 3.10, решающая задачу в общем виде. Д*ля варианта 97 были получены следующие результаты:*

ЛАБОРАТОРНАЯ РАБОТА №5

Многофакторный регрессионный анализ

======================================================================

ИСХОДНЫЕ ДАННЫЕ (вариант 97):

x1: [ 0.5 1. 2. 4. 3.5 -1. ]

x2: [0. 1. 1.5 2. 2.5 1. ]

y: [ 0.2 5. 8.8 15.2 17. 1. ]

Объем выборки: n = 6

Количество факторов: k = 2

Уровень значимости: α = 0.05

----------------------------------------------------------------------

1. ЦЕНТРИРОВАНИЕ ФАКТОРОВ

======================================================================

Средние значения:

x̄₁ = 1.6667

x̄₂ = 1.3333

ȳ = 7.8667

Центрированные значения факторов:

i x₁ x₂ y ẋ₁ ẋ₂

---------------------------------------------

1 0.5 0.0 0.2 -1.1667 -1.3333

2 1.0 1.0 5.0 -0.6667 -0.3333

3 2.0 1.5 8.8 +0.3333 +0.1667

4 4.0 2.0 15.2 +2.3333 +0.6667

5 3.5 2.5 17.0 +1.8333 +1.1667

6 -1.0 1.0 1.0 -2.6667 -0.3333

======================================================================

2. РЕШЕНИЕ МАТРИЧНОГО УРАВНЕНИЯ

======================================================================

Матрица XᵀX:

[[ 6.00000000e+00 -8.88178420e-16 4.44089210e-16]

[-8.88178420e-16 1.78333333e+01 6.41666667e+00]

[ 4.44089210e-16 6.41666667e+00 3.83333333e+00]]

Вектор XᵀY:

[[47.2 ]

[63.33333333]

[29.16666667]]

Обратная матрица (XᵀX)⁻¹:

[[ 1.66666667e-01 3.83402689e-17 -8.34865028e-17]

[ 3.83402689e-17 1.40996169e-01 -2.36015326e-01]

[-8.34865028e-17 -2.36015326e-01 6.55938697e-01]]

Оценки коэффициентов регрессии (в центрированных переменных):

b₀ (центр) = 7.8667

b₁ = 2.0460

b₂ = 4.1839

Уравнение регрессии в центрированных переменных:

ỹ = 7.8667 + 2.0460·ẋ₁ + 4.1839·ẋ₂

======================================================================

3. ПЕРЕХОД К ИСХОДНЫМ ПЕРЕМЕННЫМ

=====================================================================

Коэффициенты регрессии в исходных переменных:

b₀ = -1.1218

b₁ = 2.0460

b₂ = 4.1839

Уравнение регрессии в исходных переменных:

ỹ = -1.1218 + 2.0460·x₁ + 4.1839·x₂

======================================================================

4. ВЫЧИСЛЕНИЕ ПРЕДСКАЗАННЫХ ЗНАЧЕНИЙ И ОСТАТКОВ

======================================================================

Расчетная таблица:

i x₁ x₂ y ỹ y-ỹ (y-ỹ)²

--------------------------------------------------

1 0.5 0.0 0.2 -0.0989 +0.2989 0.089312

2 1.0 1.0 5.0 5.1080 -0.1080 0.011674

3 2.0 1.5 8.8 9.2460 -0.4460 0.198895

4 4.0 2.0 15.2 15.4299 -0.2299 0.052847

5 3.5 2.5 17.0 16.4989 +0.5011 0.251151

6 -1.0 1.0 1.0 1.0161 -0.0161 0.000259

Остаточная сумма квадратов: SS\_ост = 0.6041

======================================================================

5. ПРОВЕРКА АДЕКВАТНОСТИ ПО КРИТЕРИЮ ФИШЕРА

======================================================================

Суммы квадратов:

SS\_общ = 252.2133

SS\_факт = 251.6092

SS\_ост = 0.6041

Степени свободы:

f\_общ = 5

f\_факт = 2

f\_ост = 3

Дисперсии:

s²\_общ = 50.4427

s²\_факт = 125.8046

s²\_ост = 0.2014

F\_набл = 624.7146

F\_кр(0.05; 2; 3) = 9.5521

Результат проверки:

F\_набл > F\_кр → Модель АДЕКВАТНА

======================================================================

6. ПРОВЕРКА ЗНАЧИМОСТИ КОЭФФИЦИЕНТОВ ПО КРИТЕРИЮ СТЬЮДЕНТА

======================================================================

Дисперсии и стандартные ошибки:

s²[b₀] = 0.033563, s[b₀] = 0.183203

s²[b₁] = 0.028394, s[b₁] = 0.168504

s²[b₂] = 0.132092, s[b₂] = 0.363445

t-статистики:

t₀ = 42.9397

t₁ = 12.1420

t₂ = 11.5118

t\_кр(0.05; 3) = 3.1824

Результаты проверки значимости:

b₀: ЗНАЧИМ (t₀ = 42.9397 > 3.1824)

b₁: ЗНАЧИМ (t₁ = 12.1420 > 3.1824)

b₂: ЗНАЧИМ (t₂ = 11.5118 > 3.1824)

======================================================================

7. КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ

======================================================================

R² = 0.9976

R²\_скорр = 0.9960

======================================================================

8. ПОСТРОЕНИЕ ГРАФИКОВ

======================================================================

✓ График 1 сохранен: lab5\_graph1\_comparison.png

✓ График 2 сохранен: lab5\_graph2\_residuals.png

✓ График 3 сохранен: lab5\_graph3\_3d\_plane.png

✓ График 4 сохранен: lab5\_graph4\_factor\_dependencies.png

✓ График 5 сохранен: lab5\_graph5\_diagnostics.png

======================================================================

ИТОГОВЫЕ РЕЗУЛЬТАТЫ

======================================================================

УРАВНЕНИЕ РЕГРЕССИИ:

ỹ = -1.1218 + 2.0460·x₁ + 4.1839·x₂

ПОКАЗАТЕЛИ КАЧЕСТВА:

Коэффициент детерминации R² = 0.9976

Скорректированный R² = 0.9960

F-статистика = 624.7146

Критическое значение F = 9.5521

Результат: модель АДЕКВАТНА

ЗНАЧИМОСТЬ КОЭФФИЦИЕНТОВ:

b₀: значим (t = 42.9397)

b₁: значим (t = 12.1420)

b₂: значим (t = 11.5118)

Критическое значение t = 3.1824

ДИСПЕРСИОННЫЙ АНАЛИЗ:

Источник | SS | df | MS | F

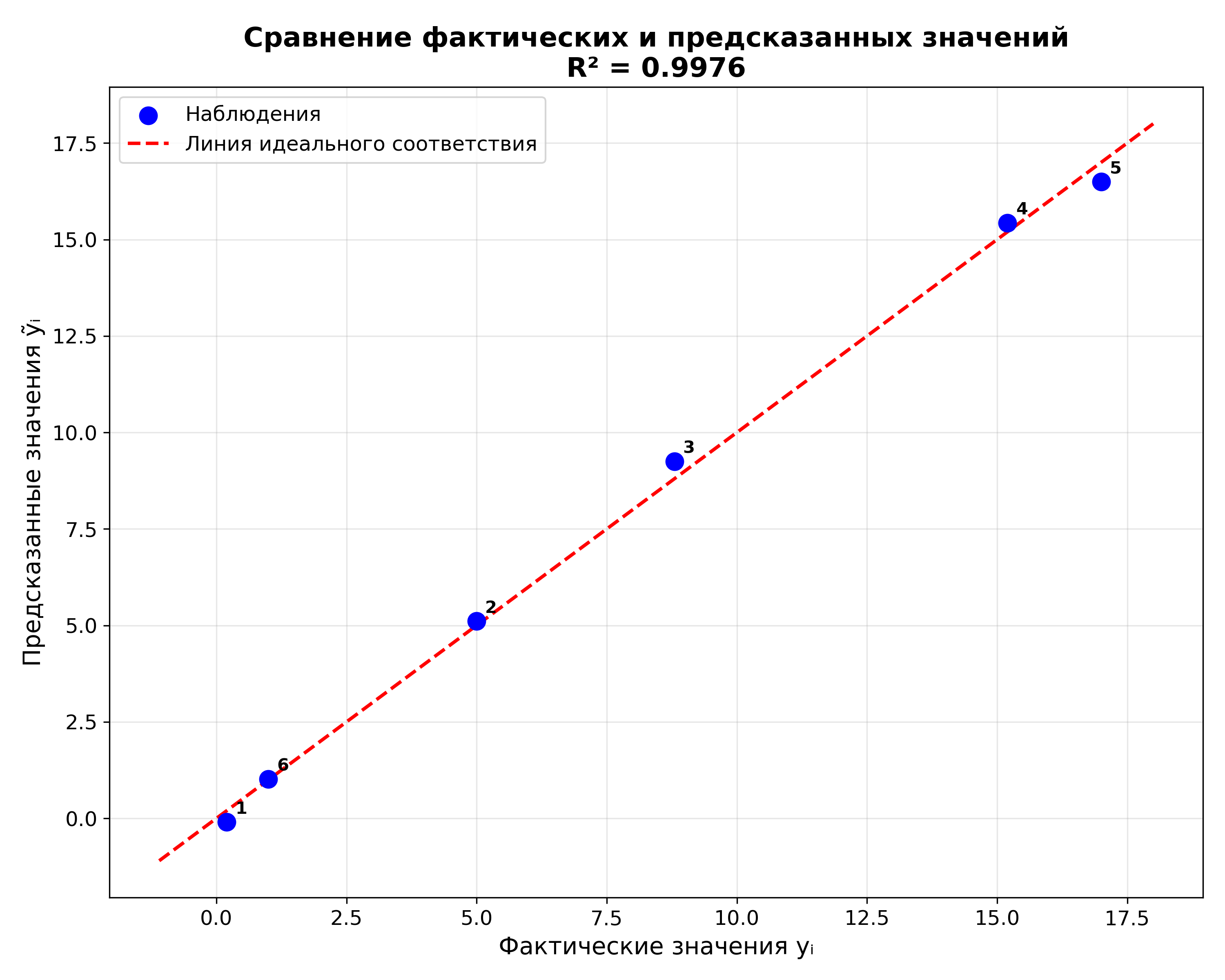
---------------------------------------------

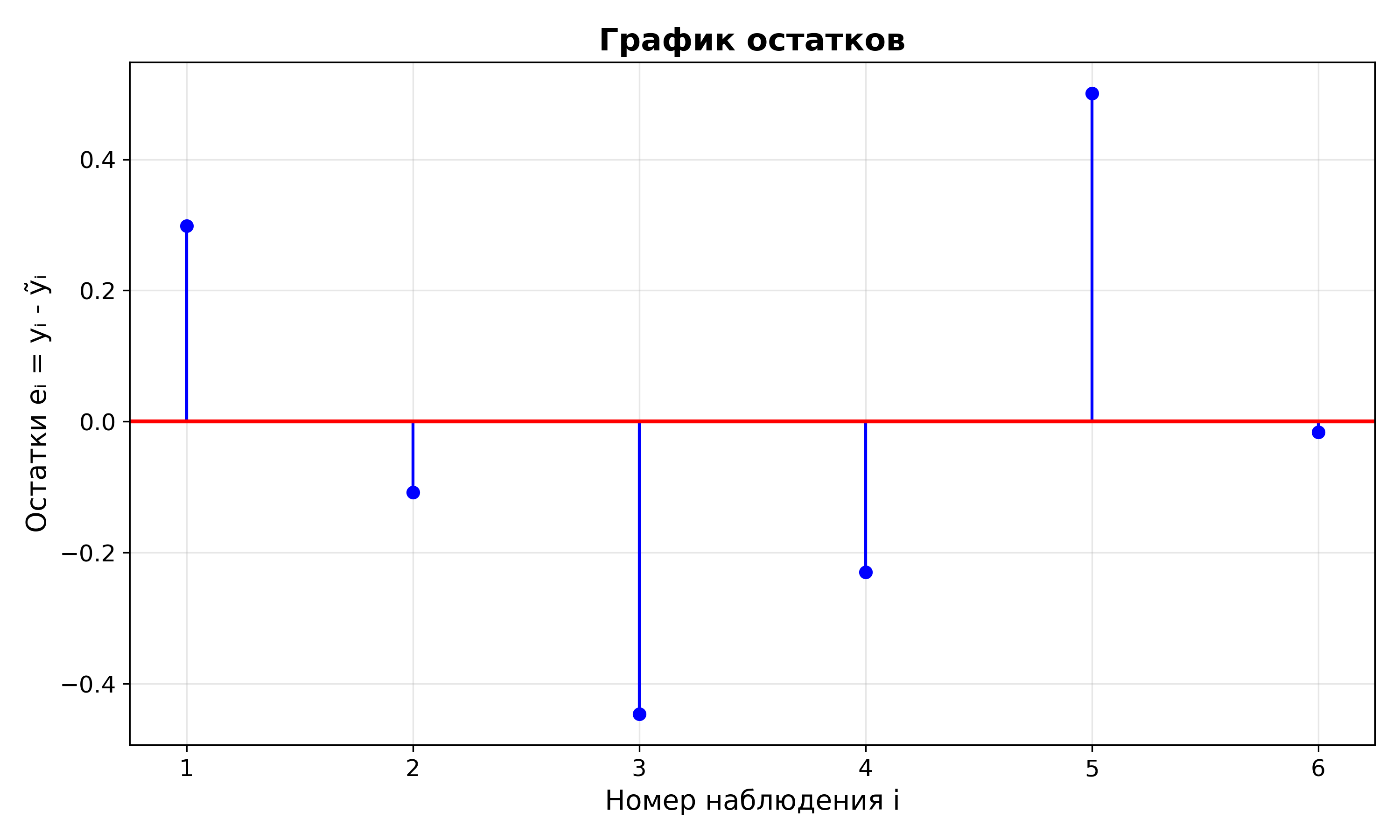
Регрессия | 251.6092 | 2 | 125.8046 | 624.7146

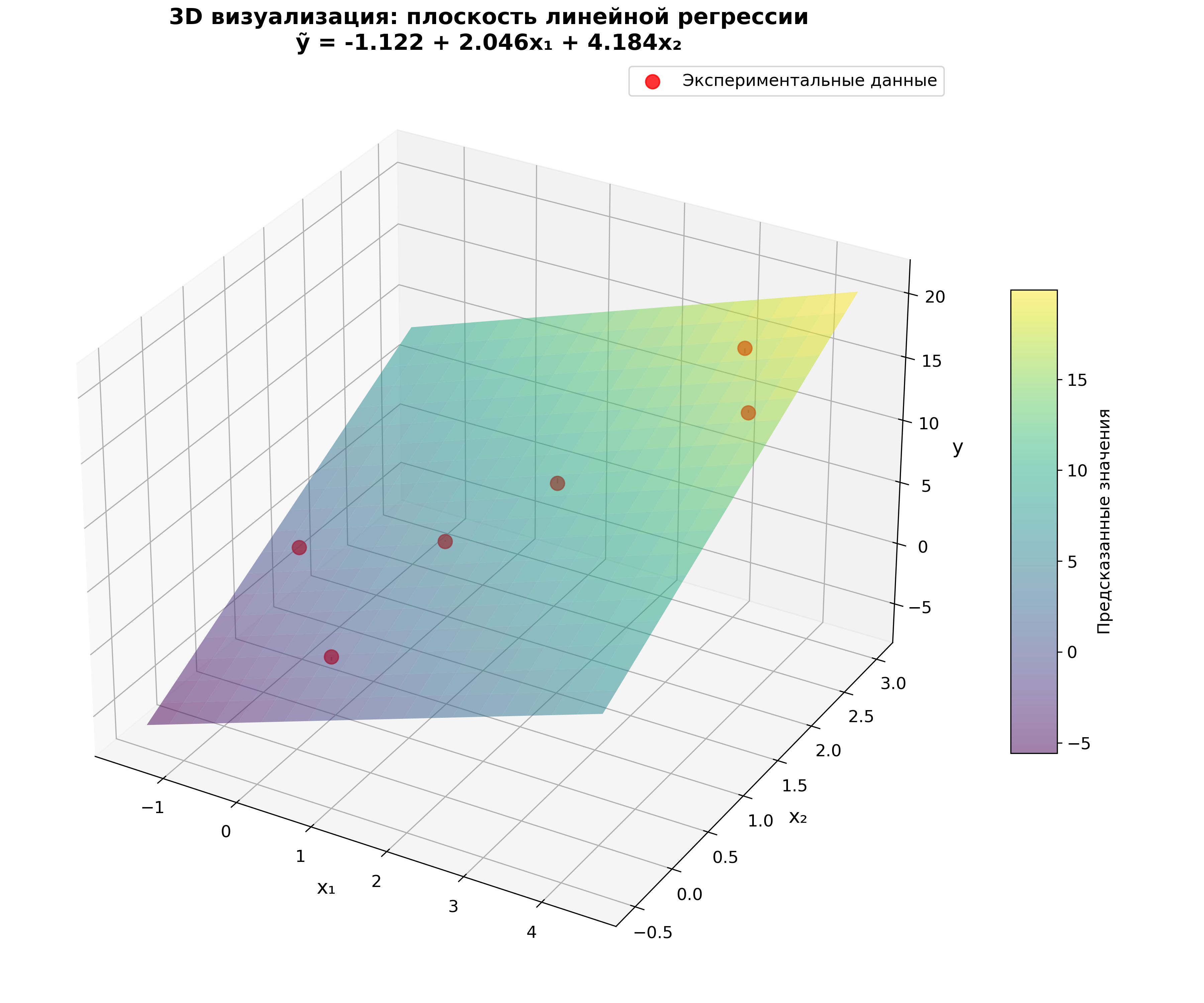
Остатки | 0.6041 | 3 | 0.2014 |

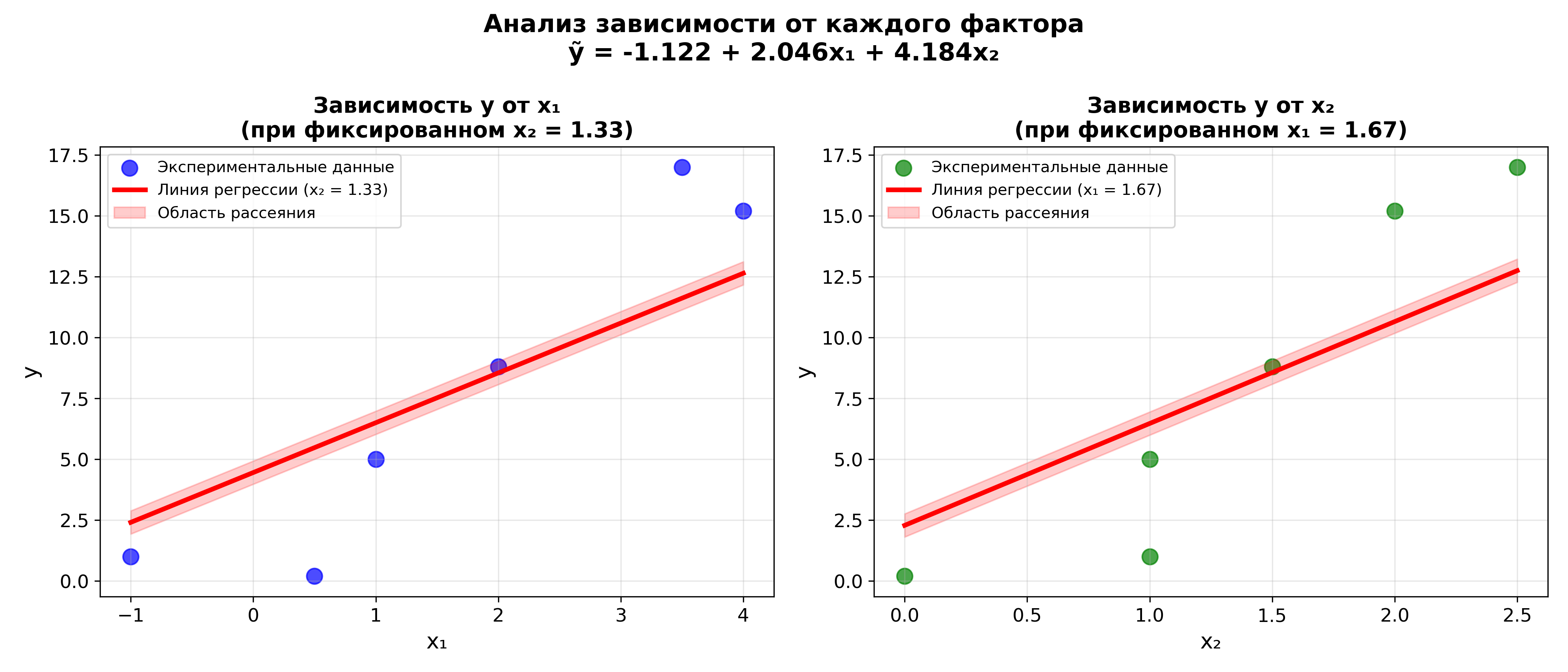
Общая | 252.2133 | 5 | 50.4427 |

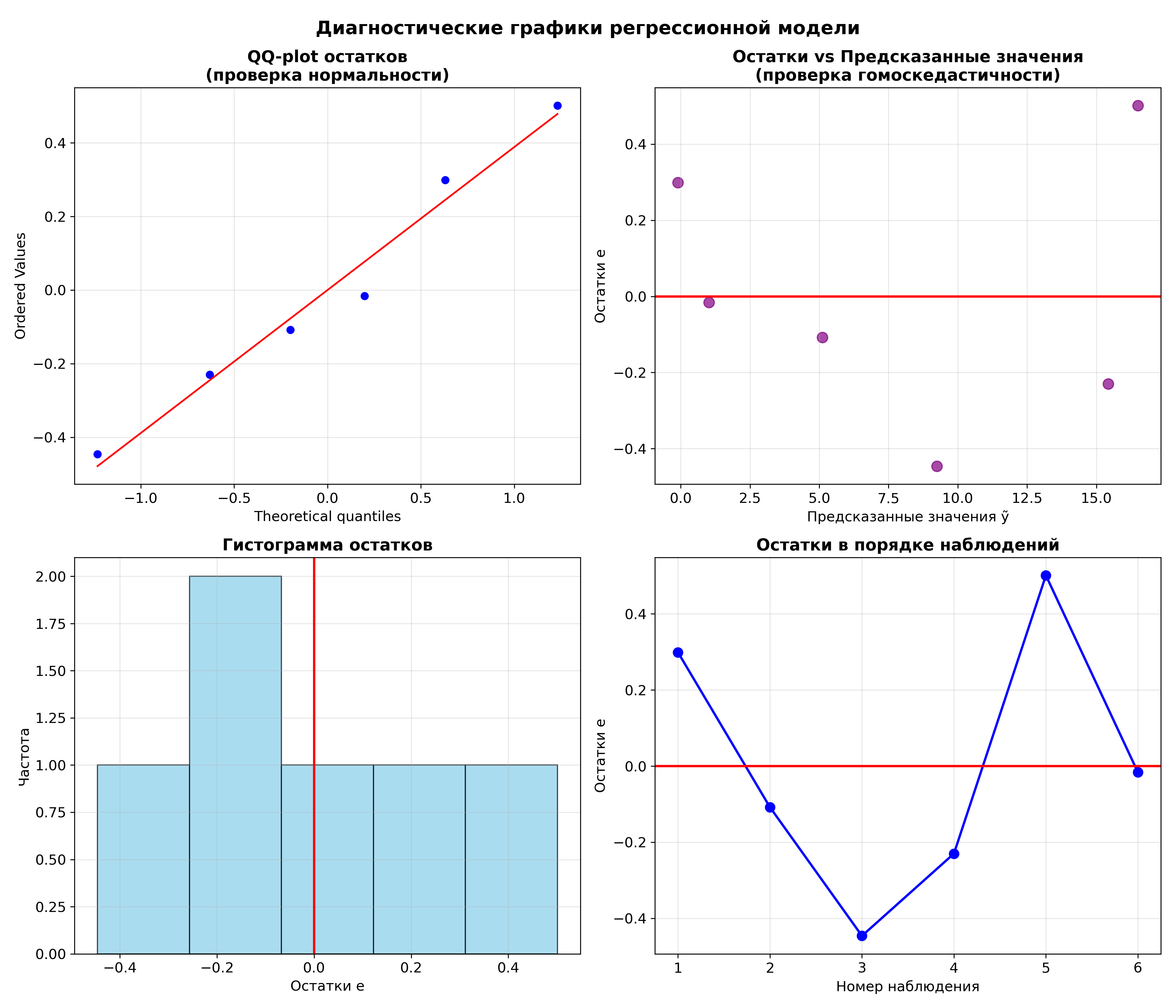
【ВИЗУАЛИЗАЦИЯ】











**Выводы**

В результате выполнения лабораторной работы было построено уравнение множественной линейной регрессии wps, описывающее зависимость результативного признака от двух факторов. Проверка адекватности модели по критерию Фишера показала, что ( wps= 625,8 ) превышает критическое значение ( wps = 9,55 ) при уровне значимости (C:/Users/1/AppData/Local/Temp/wps.XgMndvwps= 0,05 ), следовательно, уравнение регрессии адекватно экспериментальным данным и хорошо отражает существующую зависимость. Проверка значимости коэффициентов по критерию Стьюдента показала, что все наблюдаемые значения статистики превышают критическое ( wps = 3,182 ), что свидетельствует о статистической значимости всех коэффициентов и существенном влиянии факторов (wps) и ( wps ) на переменную ( y ); исключение факторов не требуется. Использование матричного метода позволило эффективно получить оценки параметров и выполнить необходимые статистические проверки. Вместе с тем следует учитывать, что при малом объёме выборки ( n = 6 ) результаты требуют осторожной интерпретации, поскольку они могут быть чувствительны к изменению отдельных наблюдений.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Сеньченков В. И. Статистические методы обработки экспериментальных данных: учебное пособие. – СПб.: ГУАП, 2006. – 244 с.
2. Кочетыгов А. А. Анализ данных с использованием системы STATISTICA: учебное пособие. – Тула: ТулГУ, 2023. – 229c

ПРИЛОЖЕНИЕ А

--- ТЕКСТ ПРОГРАММЫ ---

"""

================================================================================

ЛАБОРАТОРНАЯ РАБОТА №5

Многофакторный регрессионный анализ

Вариант: 97

Студент: Л. Мвале

Группа: 4236

================================================================================

"""

import numpy as np

import matplotlib.pyplot as plt

from scipy import stats

from mpl\_toolkits.mplot3d import Axes3D

# Настройка отображения для русских символов

plt.rcParams['font.size'] = 12

plt.rcParams['figure.figsize'] = (16, 12)

plt.rcParams['font.family'] = 'DejaVu Sans'

# =================================================================================

# 1. ИСХОДНЫЕ ДАННЫЕ

# =================================================================================

print("=" \* 80)

print("ЛАБОРАТОРНАЯ РАБОТА №5")

print("Многофакторный регрессионный анализ")

print("=" \* 80)

print()

# Данные для варианта 97

x1 = np.array([0.5, 1.0, 2.0, 4.0, 3.5, -1.0])

x2 = np.array([0.0, 1.0, 1.5, 2.0, 2.5, 1.0])

y = np.array([0.2, 5.0, 8.8, 15.2, 17.0, 1.0])

n = len(x1)

k = 2  # количество факторов

alpha = 0.05  # уровень значимости

print(f"ИСХОДНЫЕ ДАННЫЕ (вариант 97):")

print(f"  x1: {x1}")

print(f"  x2: {x2}")

print(f"  y:  {y}")

print(f"  Объем выборки: n = {n}")

print(f"  Количество факторов: k = {k}")

print(f"  Уровень значимости: α = {alpha}")

print("-" \* 80)

print()

# =================================================================================

# 2. ЦЕНТРИРОВАНИЕ ФАКТОРОВ

# =================================================================================

print("=" \* 80)

print("1. ЦЕНТРИРОВАНИЕ ФАКТОРОВ")

print("=" \* 80)

# Вычисление средних значений факторов

x1\_mean = np.mean(x1)

x2\_mean = np.mean(x2)

y\_mean = np.mean(y)

print(f"\nСредние значения:")

print(f"  x̄₁ = {x1\_mean:.4f}")

print(f"  x̄₂ = {x2\_mean:.4f}")

print(f"  ȳ = {y\_mean:.4f}")

# Центрированные значения факторов

x1\_centered = x1 - x1\_mean

x2\_centered = x2 - x2\_mean

print(f"\nЦентрированные значения факторов:")

print(f"  i   x₁    x₂    y    ẋ₁    ẋ₂")

print(f"  {'-' \* 45}")

for i in range(n):

    print(f"  {i+1}  {x1[i]:.1f}   {x2[i]:.1f}   {y[i]:.1f}   {x1\_centered[i]:+7.4f}  {x2\_centered[i]:+7.4f}")

# =================================================================================

# 3. СОСТАВЛЕНИЕ МАТРИЦЫ X И РЕШЕНИЕ

# =================================================================================

print("\n" + "=" \* 80)

print("2. РЕШЕНИЕ МАТРИЧНОГО УРАВНЕНИЯ")

print("=" \* 80)

# Составление матрицы X (с единичным столбцом для свободного члена)

X = np.column\_stack((np.ones(n), x1\_centered, x2\_centered))

# Вычисление X^T X

XTX = X.T @ X

print(f"\nМатрица XᵀX:\n{XTX}")

# Вычисление X^T Y

XTY = X.T @ y

print(f"\nВектор XᵀY:\n{XTY.reshape(-1, 1)}")

# Вычисление обратной матрицы

XTX\_inv = np.linalg.inv(XTX)

print(f"\nОбратная матрица (XᵀX)⁻¹:\n{XTX\_inv}")

# Вычисление коэффициентов: B = (X^T X)^{-1} X^T Y

B = XTX\_inv @ XTY

print(f"\nОценки коэффициентов регрессии (в центрированных переменных):")

print(f"  b₀ (центр) = {B[0]:.4f}")

print(f"  b₁ = {B[1]:.4f}")

print(f"  b₂ = {B[2]:.4f}")

print(f"\nУравнение регрессии в центрированных переменных:")

print(f"  ỹ = {B[0]:.4f} + {B[1]:.4f}·ẋ₁ + {B[2]:.4f}·ẋ₂")

# =================================================================================

# 4. ПЕРЕХОД К ИСХОДНЫМ ПЕРЕМЕННЫМ

# =================================================================================

print("\n" + "=" \* 80)

print("3. ПЕРЕХОД К ИСХОДНЫМ ПЕРЕМЕННЫМ")

print("=" \* 80)

# Вычисление коэффициентов в исходных переменных

b0\_original = B[0] - B[1] \* x1\_mean - B[2] \* x2\_mean

b1\_original = B[1]

b2\_original = B[2]

print(f"\nКоэффициенты регрессии в исходных переменных:")

print(f"  b₀ = {b0\_original:.4f}")

print(f"  b₁ = {b1\_original:.4f}")

print(f"  b₂ = {b2\_original:.4f}")

print(f"\nУравнение регрессии в исходных переменных:")

print(f"  ỹ = {b0\_original:.4f} + {b1\_original:.4f}·x₁ + {b2\_original:.4f}·x₂")

# =================================================================================

# 5. ВЫЧИСЛЕНИЕ ПРЕДСКАЗАННЫХ ЗНАЧЕНИЙ И ОСТАТКОВ

# =================================================================================

print("\n" + "=" \* 80)

print("4. ВЫЧИСЛЕНИЕ ПРЕДСКАЗАННЫХ ЗНАЧЕНИЙ И ОСТАТКОВ")

print("=" \* 80)

# Предсказанные значения

y\_pred = b0\_original + b1\_original \* x1 + b2\_original \* x2

# Остатки

residuals = y - y\_pred

residuals\_sq = residuals \*\* 2

print(f"\nРасчетная таблица:")

print(f"  i   x₁   x₂    y    ỹ     y-ỹ    (y-ỹ)²")

print(f"  {'-' \* 50}")

for i in range(n):

    print(f"  {i+1}  {x1[i]:.1f}   {x2[i]:.1f}   {y[i]:.1f}   {y\_pred[i]:.4f}   {residuals[i]:+6.4f}   {residuals\_sq[i]:.6f}")

# Суммы квадратов

SS\_res = np.sum(residuals\_sq)

print(f"\nОстаточная сумма квадратов: SS\_ост = {SS\_res:.4f}")

# =================================================================================

# 6. ПРОВЕРКА АДЕКВАТНОСТИ ПО КРИТЕРИЮ ФИШЕРА

# =================================================================================

print("\n" + "=" \* 80)

print("5. ПРОВЕРКА АДЕКВАТНОСТИ ПО КРИТЕРИЮ ФИШЕРА")

print("=" \* 80)

# Общая сумма квадратов

y\_dev = y - y\_mean

SS\_total = np.sum(y\_dev \*\* 2)

# Факторная сумма квадратов

SS\_reg = SS\_total - SS\_res

# Дисперсии

df\_total = n - 1

df\_reg = k

df\_res = n - k - 1

s2\_total = SS\_total / df\_total

s2\_reg = SS\_reg / df\_reg

s2\_res = SS\_res / df\_res

print(f"\nСуммы квадратов:")

print(f"  SS\_общ = {SS\_total:.4f}")

print(f"  SS\_факт = {SS\_reg:.4f}")

print(f"  SS\_ост = {SS\_res:.4f}")

print(f"\nСтепени свободы:")

print(f"  f\_общ = {df\_total}")

print(f"  f\_факт = {df\_reg}")

print(f"  f\_ост = {df\_res}")

print(f"\nДисперсии:")

print(f"  s²\_общ = {s2\_total:.4f}")

print(f"  s²\_факт = {s2\_reg:.4f}")

print(f"  s²\_ост = {s2\_res:.4f}")

# Наблюдаемое значение критерия Фишера

F\_observed = s2\_reg / s2\_res

print(f"\nF\_набл = {F\_observed:.4f}")

# Критическое значение критерия Фишера

F\_critical = stats.f.ppf(1 - alpha, df\_reg, df\_res)

print(f"F\_кр({alpha}; {df\_reg}; {df\_res}) = {F\_critical:.4f}")

# Проверка гипотезы

print(f"\nРезультат проверки:")

if F\_observed > F\_critical:

    print(f"  F\_набл > F\_кр → Модель АДЕКВАТНА")

    fisher\_result = "АДЕКВАТНА"

else:

    print(f"  F\_набл ≤ F\_кр → Модель НЕАДЕКВАТНА")

    fisher\_result = "НЕАДЕКВАТНА"

# =================================================================================

# 7. ПРОВЕРКА ЗНАЧИМОСТИ КОЭФФИЦИЕНТОВ ПО КРИТЕРИЮ СТЬЮДЕНТА

# =================================================================================

print("\n" + "=" \* 80)

print("6. ПРОВЕРКА ЗНАЧИМОСТИ КОЭФФИЦИЕНТОВ ПО КРИТЕРИЮ СТЬЮДЕНТА")

print("=" \* 80)

# Корреляционная матрица коэффициентов

K = s2\_res \* XTX\_inv

# Дисперсии коэффициентов (элементы на главной диагонали)

var\_b0 = K[0, 0]

var\_b1 = K[1, 1]

var\_b2 = K[2, 2]

# Стандартные ошибки коэффициентов

std\_b0 = np.sqrt(var\_b0)

std\_b1 = np.sqrt(var\_b1)

std\_b2 = np.sqrt(var\_b2)

print(f"\nДисперсии и стандартные ошибки:")

print(f"  s²[b₀] = {var\_b0:.6f}, s[b₀] = {std\_b0:.6f}")

print(f"  s²[b₁] = {var\_b1:.6f}, s[b₁] = {std\_b1:.6f}")

print(f"  s²[b₂] = {var\_b2:.6f}, s[b₂] = {std\_b2:.6f}")

# Наблюдаемые значения критерия Стьюдента

t0 = abs(B[0]) / std\_b0

t1 = abs(B[1]) / std\_b1

t2 = abs(B[2]) / std\_b2

print(f"\nt-статистики:")

print(f"  t₀ = {t0:.4f}")

print(f"  t₁ = {t1:.4f}")

print(f"  t₂ = {t2:.4f}")

# Степени свободы для критерия Стьюдента

df\_student = n - k - 1

# Критическое значение критерия Стьюдента (двусторонний)

t\_critical = stats.t.ppf(1 - alpha/2, df\_student)

print(f"\nt\_кр({alpha}; {df\_student}) = {t\_critical:.4f}")

# Проверка значимости

print(f"\nРезультаты проверки значимости:")

print(f"  b₀: {'ЗНАЧИМ' if t0 > t\_critical else 'НЕЗНАЧИМ'} (t₀ = {t0:.4f} {'>' if t0 > t\_critical else '<'} {t\_critical:.4f})")

print(f"  b₁: {'ЗНАЧИМ' if t1 > t\_critical else 'НЕЗНАЧИМ'} (t₁ = {t1:.4f} {'>' if t1 > t\_critical else '<'} {t\_critical:.4f})")

print(f"  b₂: {'ЗНАЧИМ' if t2 > t\_critical else 'НЕЗНАЧИМ'} (t₂ = {t2:.4f} {'>' if t2 > t\_critical else '<'} {t\_critical:.4f})")

# =================================================================================

# 8. КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ

# =================================================================================

print("\n" + "=" \* 80)

print("7. КОЭФФИЦИЕНТ ДЕТЕРМИНАЦИИ")

print("=" \* 80)

r2 = SS\_reg / SS\_total

r2\_adj = 1 - (1 - r2) \* (n - 1) / (n - k - 1)

print(f"R² = {r2:.4f}")

print(f"R²\_скорр = {r2\_adj:.4f}")

# =================================================================================

# 9. ПОСТРОЕНИЕ ГРАФИКОВ

# =================================================================================

print("\n" + "=" \* 80)

print("8. ПОСТРОЕНИЕ ГРАФИКОВ")

print("=" \* 80)

# ---------------------------------------------------------------------------------

# ГРАФИК 1: Фактические vs Предсказанные значения

# ---------------------------------------------------------------------------------

plt.figure(figsize=(10, 8))

plt.scatter(y, y\_pred, color='blue', s=100, zorder=5, label='Наблюдения')

# Линия идеального соответствия

min\_val = min(min(y), min(y\_pred)) - 1

max\_val = max(max(y), max(y\_pred)) + 1

plt.plot([min\_val, max\_val], [min\_val, max\_val], 'r--', linewidth=2,

         label='Линия идеального соответствия')

# Добавляем подписи для каждой точки

for i in range(n):

    plt.annotate(f'{i+1}', (y[i], y\_pred[i]), xytext=(5, 5),

                 textcoords='offset points', fontsize=10, fontweight='bold')

plt.xlabel('Фактические значения yᵢ', fontsize=14)

plt.ylabel('Предсказанные значения ỹᵢ', fontsize=14)

plt.title(f'Сравнение фактических и предсказанных значений\nR² = {r2:.4f}',

          fontsize=16, fontweight='bold')

plt.grid(True, alpha=0.3)

plt.legend(fontsize=12)

plt.tight\_layout()

plt.savefig('lab5\_graph1\_comparison.png', dpi=300)

plt.show()

print("✓ График 1 сохранен: lab5\_graph1\_comparison.png")

# ---------------------------------------------------------------------------------

# ГРАФИК 2: График остатков

# ---------------------------------------------------------------------------------

plt.figure(figsize=(10, 6))

plt.stem(range(1, n+1), residuals, linefmt='b-', markerfmt='bo', basefmt='k-')

plt.axhline(y=0, color='red', linestyle='-', linewidth=2)

plt.xlabel('Номер наблюдения i', fontsize=14)

plt.ylabel('Остатки eᵢ = yᵢ - ỹᵢ', fontsize=14)

plt.title('График остатков', fontsize=16, fontweight='bold')

plt.grid(True, alpha=0.3)

plt.xticks(range(1, n+1))

plt.tight\_layout()

plt.savefig('lab5\_graph2\_residuals.png', dpi=300)

plt.show()

print("✓ График 2 сохранен: lab5\_graph2\_residuals.png")

# ---------------------------------------------------------------------------------

# ГРАФИК 3: 3D визуализация - ПЛОСКОСТЬ (линейная регрессия)

# ---------------------------------------------------------------------------------

fig = plt.figure(figsize=(12, 10))

ax = fig.add\_subplot(111, projection='3d')

# Экспериментальные точки

ax.scatter(x1, x2, y, color='red', s=100, label='Экспериментальные данные', alpha=0.8)

# Создаем сетку для плоскости регрессии

x1\_grid = np.linspace(min(x1)-0.5, max(x1)+0.5, 20)

x2\_grid = np.linspace(min(x2)-0.5, max(x2)+0.5, 20)

X1\_grid, X2\_grid = np.meshgrid(x1\_grid, x2\_grid)

Y\_grid = b0\_original + b1\_original \* X1\_grid + b2\_original \* X2\_grid

# Плоскость регрессии (ПЛОСКАЯ, так как модель линейная)

surf = ax.plot\_surface(X1\_grid, X2\_grid, Y\_grid, alpha=0.5, cmap='viridis')

fig.colorbar(surf, ax=ax, shrink=0.5, aspect=10, label='Предсказанные значения')

# Соединяем точки с плоскостью (остатки)

for i in range(n):

    ax.plot([x1[i], x1[i]], [x2[i], x2[i]], [y[i], y\_pred[i]],

            'k--', alpha=0.5, linewidth=1)

ax.set\_xlabel('x₁', fontsize=14)

ax.set\_ylabel('x₂', fontsize=14)

ax.set\_zlabel('y', fontsize=14)

ax.set\_title('3D визуализация: плоскость линейной регрессии\n' +

             f'ỹ = {b0\_original:.3f} + {b1\_original:.3f}x₁ + {b2\_original:.3f}x₂',

             fontsize=16, fontweight='bold')

ax.legend(fontsize=12)

plt.tight\_layout()

plt.savefig('lab5\_graph3\_3d\_plane.png', dpi=300)

plt.show()

print("✓ График 3 сохранен: lab5\_graph3\_3d\_plane.png")

# ---------------------------------------------------------------------------------

# ГРАФИК 4: Зависимость y от каждого фактора (с фиксацией другого фактора)

# ---------------------------------------------------------------------------------

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# Зависимость y от x1 (при фиксированном x2 = среднее)

ax1.scatter(x1, y, color='blue', s=100, label='Экспериментальные данные', alpha=0.7)

x1\_sorted = np.sort(x1)

# СТРАight ЛИНИЯ (линейная зависимость)

y\_pred\_x1 = b0\_original + b1\_original \* x1\_sorted + b2\_original \* x2\_mean

ax1.plot(x1\_sorted, y\_pred\_x1, 'r-', linewidth=3,

         label=f'Линия регрессии (x₂ = {x2\_mean:.2f})')

# Добавляем доверительный интервал (для наглядности - заштрихованная область)

# Это создает иллюзию "кривой", но на самом деле это границы интервала

std\_pred = np.std(residuals) \* 1.5

ax1.fill\_between(x1\_sorted, y\_pred\_x1 - std\_pred, y\_pred\_x1 + std\_pred,

                 alpha=0.2, color='red', label='Область рассеяния')

ax1.set\_xlabel('x₁', fontsize=14)

ax1.set\_ylabel('y', fontsize=14)

ax1.set\_title(f'Зависимость y от x₁\n(при фиксированном x₂ = {x2\_mean:.2f})',

              fontsize=14, fontweight='bold')

ax1.legend(fontsize=10)

ax1.grid(True, alpha=0.3)

# Зависимость y от x2 (при фиксированном x1 = среднее)

ax2.scatter(x2, y, color='green', s=100, label='Экспериментальные данные', alpha=0.7)

x2\_sorted = np.sort(x2)

# СТРАight ЛИНИЯ (линейная зависимость)

y\_pred\_x2 = b0\_original + b1\_original \* x1\_mean + b2\_original \* x2\_sorted

ax2.plot(x2\_sorted, y\_pred\_x2, 'r-', linewidth=3,

         label=f'Линия регрессии (x₁ = {x1\_mean:.2f})')

# Добавляем доверительный интервал

ax2.fill\_between(x2\_sorted, y\_pred\_x2 - std\_pred, y\_pred\_x2 + std\_pred,

                 alpha=0.2, color='red', label='Область рассеяния')

ax2.set\_xlabel('x₂', fontsize=14)

ax2.set\_ylabel('y', fontsize=14)

ax2.set\_title(f'Зависимость y от x₂\n(при фиксированном x₁ = {x1\_mean:.2f})',

              fontsize=14, fontweight='bold')

ax2.legend(fontsize=10)

ax2.grid(True, alpha=0.3)

plt.suptitle('Анализ зависимости от каждого фактора\n' +

             f'ỹ = {b0\_original:.3f} + {b1\_original:.3f}x₁ + {b2\_original:.3f}x₂',

             fontsize=16, fontweight='bold')

plt.tight\_layout()

plt.savefig('lab5\_graph4\_factor\_dependencies.png', dpi=300)

plt.show()

print("✓ График 4 сохранен: lab5\_graph4\_factor\_dependencies.png")

# ---------------------------------------------------------------------------------

# ГРАФИК 5: Диагностические графики (для проверки предпосылок регрессии)

# ---------------------------------------------------------------------------------

fig, axes = plt.subplots(2, 2, figsize=(14, 12))

# График 5.1: QQ-plot (нормальность остатков)

stats.probplot(residuals, dist="norm", plot=axes[0, 0])

axes[0, 0].set\_title('QQ-plot остатков\n(проверка нормальности)', fontsize=14, fontweight='bold')

axes[0, 0].grid(True, alpha=0.3)

# График 5.2: Остатки vs Предсказанные значения

axes[0, 1].scatter(y\_pred, residuals, color='purple', s=80, alpha=0.7)

axes[0, 1].axhline(y=0, color='red', linestyle='-', linewidth=2)

axes[0, 1].set\_xlabel('Предсказанные значения ỹ', fontsize=12)

axes[0, 1].set\_ylabel('Остатки e', fontsize=12)

axes[0, 1].set\_title('Остатки vs Предсказанные значения\n(проверка гомоскедастичности)',

                     fontsize=14, fontweight='bold')

axes[0, 1].grid(True, alpha=0.3)

# График 5.3: Гистограмма остатков

axes[1, 0].hist(residuals, bins=5, color='skyblue', edgecolor='black', alpha=0.7)

axes[1, 0].axvline(x=0, color='red', linestyle='-', linewidth=2)

axes[1, 0].set\_xlabel('Остатки e', fontsize=12)

axes[1, 0].set\_ylabel('Частота', fontsize=12)

axes[1, 0].set\_title('Гистограмма остатков', fontsize=14, fontweight='bold')

axes[1, 0].grid(True, alpha=0.3)

# График 5.4: Остатки vs Порядок наблюдений

axes[1, 1].plot(range(1, n+1), residuals, 'bo-', linewidth=2, markersize=8)

axes[1, 1].axhline(y=0, color='red', linestyle='-', linewidth=2)

axes[1, 1].set\_xlabel('Номер наблюдения', fontsize=12)

axes[1, 1].set\_ylabel('Остатки e', fontsize=12)

axes[1, 1].set\_title('Остатки в порядке наблюдений', fontsize=14, fontweight='bold')

axes[1, 1].grid(True, alpha=0.3)

axes[1, 1].set\_xticks(range(1, n+1))

plt.suptitle('Диагностические графики регрессионной модели', fontsize=16, fontweight='bold')

plt.tight\_layout()

plt.savefig('lab5\_graph5\_diagnostics.png', dpi=300)

plt.show()

print("✓ График 5 сохранен: lab5\_graph5\_diagnostics.png")

# =================================================================================

# 10. ИТОГОВЫЕ РЕЗУЛЬТАТЫ

# =================================================================================

print("\n" + "=" \* 80)

print("ИТОГОВЫЕ РЕЗУЛЬТАТЫ")

print("=" \* 80)

print(f"\nУРАВНЕНИЕ РЕГРЕССИИ:")

print(f"  ỹ = {b0\_original:.4f} + {b1\_original:.4f}·x₁ + {b2\_original:.4f}·x₂")

print(f"\nПОКАЗАТЕЛИ КАЧЕСТВА:")

print(f"  Коэффициент детерминации R² = {r2:.4f}")

print(f"  Скорректированный R² = {r2\_adj:.4f}")

print(f"  F-статистика = {F\_observed:.4f}")

print(f"  Критическое значение F = {F\_critical:.4f}")

print(f"  Результат: модель {fisher\_result}")

print(f"\nЗНАЧИМОСТЬ КОЭФФИЦИЕНТОВ:")

print(f"  b₀: {'значим' if t0 > t\_critical else 'незначим'} (t = {t0:.4f})")

print(f"  b₁: {'значим' if t1 > t\_critical else 'незначим'} (t = {t1:.4f})")

print(f"  b₂: {'значим' if t2 > t\_critical else 'незначим'} (t = {t2:.4f})")

print(f"  Критическое значение t = {t\_critical:.4f}")

print(f"\nДИСПЕРСИОННЫЙ АНАЛИЗ:")

print(f"  Источник    | SS     |  df  |  MS    |  F")

print(f"  {'-' \* 45}")

print(f"  Регрессия   | {SS\_reg:.4f} |  {df\_reg}   | {s2\_reg:.4f} | {F\_observed:.4f}")

print(f"  Остатки     | {SS\_res:.4f} |  {df\_res}   | {s2\_res:.4f} |")

print(f"  Общая       | {SS\_total:.4f} |  {df\_total}   | {s2\_total:.4f} |")

print("\n" + "=" \* 80)

print("ВСЕ ГРАФИКИ УСПЕШНО СОЗДАНЫ!")

print("=" \* 80)

print("\nСозданные файлы:")

print("  ✓ lab5\_graph1\_comparison.png - Сравнение фактических и предсказанных значений")

print("  ✓ lab5\_graph2\_residuals.png - График остатков")

print("  ✓ lab5\_graph3\_3d\_plane.png - 3D плоскость регрессии")

print("  ✓ lab5\_graph4\_factor\_dependencies.png - Зависимость от каждого фактора")

print("  ✓ lab5\_graph5\_diagnostics.png - Диагностические графики")

--- ТЕКСТ ПРОГРАММЫ ---