ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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

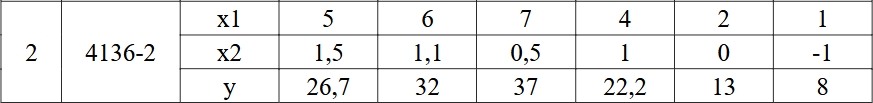
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  |  |  | Н.С. Бобрович |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

1. **Цель работы:**

Цель данной работы заключается в разработке и анализе линейной регрессионной модели на основе предоставленных экспериментальных данных. Основные задачи включают центрирование факторов, составление матричного уравнения, оценку коэффициентов регрессии методом наименьших квадратов, проверку адекватности модели по критерию Фишера и проведение селекции факторов по критерию Стьюдента. В результате была построена модель, которая описывает зависимость выходной величины от двух входных факторов, а также проведена оценка значимости факторов и их вклада в общую вариацию результата.

1. **Задание на лабораторную работу:**

Скриншот 17-02-2025 201039

Порядок выполнения задания:

1. Выполнить центрирование факторов (массив экспериментальных данных;

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;

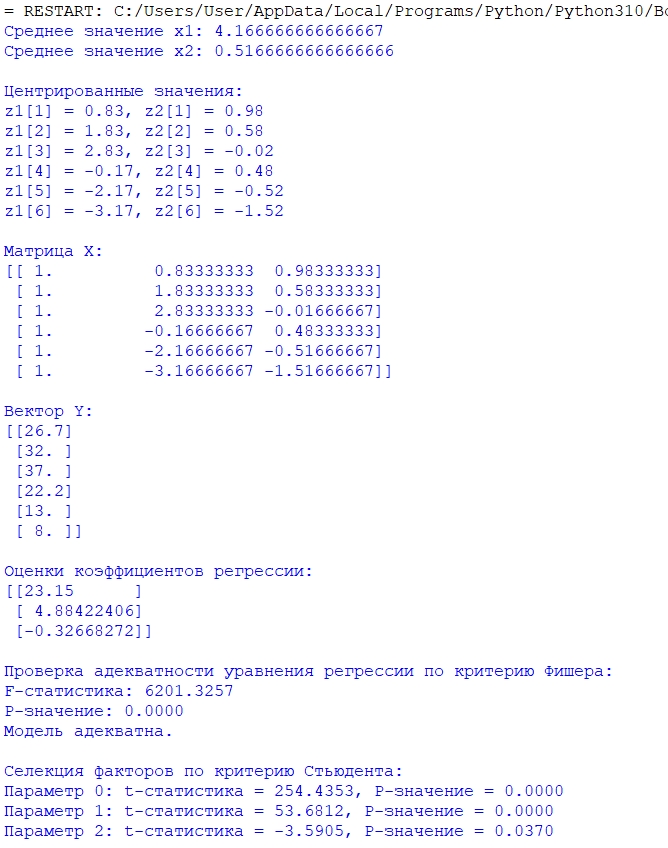
5. Выполнить селекцию факторов по критерию Стьюдента при таком же уровне значимости;

6. Повторно проверить адекватность уравнения регрессии после исключения незначимых факторов.

1. **Ход работы:**

Код на Python в приложении 1.

1. **Результат работы:**



1. **Выводы:**
2. Центрирование факторов: Данные были приведены к нулевому среднему значению, что позволяет упростить дальнейшие расчеты и интерпретацию результатов.
3. Матричная форма уравнения регрессии: Было составлено матричное уравнение для нахождения оценок коэффициентов регрессии методом наименьших квадратов.
4. Оценка коэффициентов регрессии: Решая матричное уравнение, были найдены оценки коэффициентов регрессии. Это позволило построить линейную модель зависимости отклика от факторов.
5. Адекватность модели: Модель была признана адекватной по критерию Фишера, так как F-статистика оказалась достаточно высокой, а соответствующее P-значение оказалось меньше уровня значимости α = 0.05.
6. Селекция факторов по критерию Стьюдента: Была проведена оценка значимости каждого коэффициента с помощью t-критерия Стьюдента. Незначимых факторов не оказалось.
7. Повторная проверка адекватности: Так как незначимых факторов не оказалось, повторная роверка не нужна, модель адекватна.

Проведенный анализ позволил создать эффективную регрессионную модель, способную предсказывать выходные значения на основе входных факторов с учетом их значимости и влияния.

1. **Список литературы:**
2. Сеньченков В.И.: Статистические методы обработки экспериментальных данных. - 191 стр. - Санкт-Петербург - 2006 г.
3. Мышко В.В.: Лекция 1. - 45 стр. - Санкт-Петербург - 2025 г.
4. **Приложения:**

Приложение 1:

import numpy as np

from scipy import stats

# Исходные данные

x1 = np.array([5, 6, 7, 4, 2, 1])

x2 = np.array([1.5, 1.1, 0.5, 1, 0, -1])

y = np.array([26.7, 32, 37, 22.2, 13, 8])

# Шаг 1: Центрирование факторов

mean\_x1 = np.mean(x1)

mean\_x2 = np.mean(x2)

print("Среднее значение x1:", mean\_x1)

print("Среднее значение x2:", mean\_x2)

z1 = x1 - mean\_x1

z2 = x2 - mean\_x2

print("\nЦентрированные значения:")

for i in range(len(z1)):

print(f"z1[{i+1}] = {z1[i]:.2f}, z2[{i+1}] = {z2[i]:.2f}")

# Шаг 2: Построение матрицы X и вектора Y

X = np.column\_stack((np.ones\_like(y), z1, z2))

Y = y.reshape(-1, 1)

print("\nМатрица X:")

print(X)

print("\nВектор Y:")

print(Y)

# Шаг 3: Нахождение оценок коэффициентов регрессии

XT\_X = np.dot(X.T, X)

inv\_XT\_X = np.linalg.inv(XT\_X)

XT\_Y = np.dot(X.T, Y)

B = np.dot(inv\_XT\_X, XT\_Y)

print("\nОценки коэффициентов регрессии:")

print(B)

# Шаг 4: Проверка адекватности уравнения регрессии по критерию Фишера

# Рассчитаем сумму квадратов ошибок (SSE) и полную сумму квадратов (SST)

y\_pred = np.dot(X, B)

residuals = y - y\_pred.flatten()

SSE = np.sum(residuals\*\*2)

SST = np.sum((y - np.mean(y))\*\*2)

SSR = SST - SSE

df\_model = 2 # Число параметров модели без свободного члена

df\_error = len(y) - df\_model - 1

MSR = SSR / df\_model

MSE = SSE / df\_error

F\_statistic = MSR / MSE

p\_value = 1 - stats.f.cdf(F\_statistic, df\_model, df\_error)

alpha = 0.05

print("\nПроверка адекватности уравнения регрессии по критерию Фишера:")

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

print(f"P-значение: {p\_value:.4f}")

if p\_value <= alpha:

print("Модель адекватна.")

else:

print("Модель неадекватна.")

# Шаг 5: Селекция факторов по критерию Стьюдента

std\_err = np.sqrt(np.diag(MSE \* inv\_XT\_X))

t\_stats = B / std\_err

p\_values = 2 \* (1 - stats.t.cdf(abs(t\_stats), df\_error))

# Шаг 6: Повторная проверка адекватности уравнения регрессии после исключения незначимых факторов

# Здесь нужно удалить соответствующие столбцы из матрицы X и пересчитать оценки коэффициентов

print("\nСелекция факторов по критерию Стьюдента:")

for i in range(len(p\_values)):

print(f"Параметр {i}: t-статистика = {t\_stats[i][0]:.4f}, P-значение = {p\_values[i][0]:.4f}")

if p\_values[i][0] > alpha:

print(f"\tПараметр {i} исключен из модели.")