ГУАП

КАФЕДРА № 43

|  |  |  |
| --- | --- | --- |
| ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: |  |  |

ПРЕПОДАВАТЕЛЬ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| доцент, к.т.н., доцент | / |  | / |  | / | В. В. Мышко |
| (должность, учёная степень, звание) |  | (подпись) |  | (дата защиты) |  | (инициалы, фамилия) |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5

«Многофакторный

регрессионный анализ»

ПО КУРСУ: «ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ»

|  |  |  |  |
| --- | --- | --- | --- |
| РАБОТУ ВЫПОЛНИЛ СТУДЕНТ: | 4134к | / | Н.А. Костяков |
|  | (номер группы) |  | (инициалы, фамилия) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | / |  | / |  |
|  |  | (подпись студента) |  | (дата отчета) |

Санкт-Петербург 2025

**Постановка задачи**

На основе заданного массива данных:

• построить уравнение регрессии в виде линейного алгебраического полинома

от двух переменных;

• проверить адекватность уравнения регрессии;

• 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;

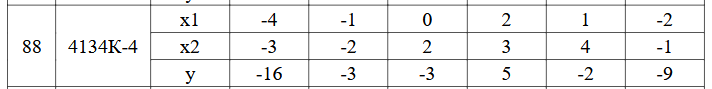
5. Выполнить селекцию факторов по критерию Стьюдента при таком же

уровне значимости;

6. Повторно проверить адекватность уравнения регрессии после исключения

незначимых факторов.

Вариант 88



**Ход выполнения**

**1. Выполнить центрирование факторов:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x1 | -3.(3) | -0.(3) | 0.(6) | 2.(6) | 1.(6) | -1.(3) |
| x2 | -3.5 | -2.5 | 1.5 | 2.5 | 3.5 | -1.5 |
| y | -11.(3) | 1.(6) | 1.(6) | 9.(6) | 2.(6) | -4(3) |

**2. Составить матричное уравнение с вектором неизвестных оценок**

**коэффициентов регрессии;**

Матричное уравнение составлю при помощи пакета Numpy

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

Y = y\_centered

Получаем такую матрицу X

|  |  |  |
| --- | --- | --- |
| 1 | -3.33333333 | -3.5 |
| 1 | -0.33333333 | -2.5 |
| 1 | 0.66666667 | 1.5 |
| 1 | 2.66666667 | 2.5 |
| 1 | 1.66666667 | 3.5 |
| 1 | 1.33333333 | -1.5 |

Вектор значений Y:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -11.33333333 | 1.66666667 | 1.66666667 | 9.66666667 | 2.66666667 | -4.33333333 |

**3. Найти оценки коэффициентов регрессии посредством решения матричного**

**уравнения;**

Коэффициенты регрессии: [1.44328993e-15, 4.88698011e+00, -1.41772152e+00]

**4. Проверить адекватность построенного уравнения регрессии**

**экспериментальным данным по критерию Фишера при уровне значимости α**

**= 0,05;**

Статистика Фишера: **884.5306**

Критическое значение F: **9.5521**

Статистика выше критического значения, следовательно **уравнение адекватно**

**5. Выполнить селекцию факторов по критерию Стьюдента при таком же**

**уровне значимости;**

t-статистики: [ 7.59456587e-15 2.21254548e+01 -8.56008787e+00]

p-значения: [1.00000000e+00 2.02120123e-04 3.35042920e-03]

Убираем из выборки факторы с p-значением более 0.05, и получаем значимые коэффициенты

[ 4.88698011, -1.41772152]

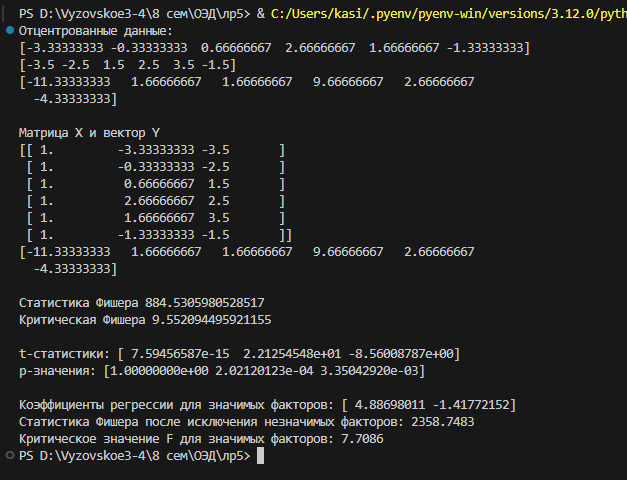
6. Повторно проверить адекватность уравнения регрессии после исключения

**незначимых факторов.**

Статистика Фишера после исключения незначимых факторов: **2358.7483**

Критическое значение F для значимых факторов: **7.7086**

**Результат работы программы на python**



**Листинг программы на python**

|  |
| --- |
| import numpy as np  import scipy.stats as stats  # Данные  x1 = np.array([-4, -1, 0, 2, 1, -2])  x2 = np.array([-3, -2, 2, 3, 4, -1])  y = np.array([-16, -3, -3, 5, -2, -9])  # Центрирование данных  x1\_centered = x1 - np.mean(x1)  x2\_centered = x2 - np.mean(x2)  y\_centered = y - np.mean(y)  print("Отцентрованные данные:")  print(x1\_centered)  print(x2\_centered)  print(y\_centered)  print()  # Составляем матрицу X с добавлением столбца для свободного члена  X = np.column\_stack((np.ones(len(x1\_centered)), x1\_centered, x2\_centered))  # столбец для 1 (свободный член)  Y = y\_centered  print("Матрица X и вектор Y")  print(X)  print(Y)  print()  # Матричное уравнение для нахождения коэффициентов  X\_transpose = X.T  beta = np.linalg.inv(X\_transpose @ X) @ X\_transpose @ Y  # Предсказания и остатки  y\_pred = X @ beta  residuals = Y - y\_pred  # Средняя квадратическая ошибка регрессии (MSR)  MSR = np.sum((y\_pred - np.mean(y))\*\*2) / (X.shape[1] - 1)  # Средняя квадратическая ошибка (MSE)  MSE = np.sum(residuals\*\*2) / (len(y) - X.shape[1])  # Статистика Фишера  F\_stat = MSR / MSE  F\_critical = stats.f.ppf(1 - 0.05, X.shape[1] - 1, len(y) - X.shape[1])  print("Статистика Фишера", F\_stat)  print("Критическая Фишера", F\_critical)  print()  # Стандартные ошибки коэффициентов  se\_beta = np.sqrt(MSE \* np.diagonal(np.linalg.inv(X\_transpose @ X)))  # t-статистики и p-значения  t\_stats = beta / se\_beta  p\_values = 2 \* (1 - stats.t.cdf(np.abs(t\_stats), df=len(y) - X.shape[1]))  # Выводим t-статистики и p-значения  print("t-статистики:", t\_stats)  print("p-значения:", p\_values)  print()  # Исключаем незначимые факторы (с p-значением > 0.05)  significant\_factors = p\_values < 0.05  X\_significant = X[:, significant\_factors]  # Пересчитываем коэффициенты с учетом значимых факторов  beta\_significant = np.linalg.inv(X\_significant.T @ X\_significant) @ X\_significant.T @ Y  # Повторная проверка адекватности после исключения незначимых факторов  y\_pred\_significant = X\_significant @ beta\_significant  residuals\_significant = Y - y\_pred\_significant  # Средняя квадратическая ошибка регрессии (MSR) для значимых факторов  MSR\_significant = np.sum((y\_pred\_significant - np.mean(y))\*\*2) / (X\_significant.shape[1] - 1)  # Средняя квадратическая ошибка (MSE) для значимых факторов  MSE\_significant = np.sum(residuals\_significant\*\*2) / (len(y) - X\_significant.shape[1])  # Статистика Фишера для значимых факторов  F\_stat\_significant = MSR\_significant / MSE\_significant  F\_critical\_significant = stats.f.ppf(1 - 0.05, X\_significant.shape[1] - 1, len(y) - X\_significant.shape[1])  # Вывод результатов  print("Коэффициенты регрессии для значимых факторов:", beta\_significant)  print(f"Статистика Фишера после исключения незначимых факторов: {F\_stat\_significant:.4f}")  print(f"Критическое значение F для значимых факторов: {F\_critical\_significant:.4f}") |