

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	/	Столяров Н.С.
(номер группы)		(инициалы, фамилия)

/	/
(подпись студента)	(дата отчета)

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 $\alpha = 0,05$;
5. Выполнить селекцию факторов по критерию Стьюдента при таком же уровне значимости;
6. Повторно проверить адекватность уравнения регрессии после исключения незначимых факторов.

Вариант 99

99	4134К-15	x1	-2	-1	0	1	2	-3
		x2	1	3	4	6	7	-10
		y	-11	-2	7	16	26	-9

Ход выполнения

Центрирование факторов:

- x_1 :
Среднее: $-2-1+0+1+2-36 = -0.5$
Центрированные значения:
 $[-1.5, -0.5, 0.5, 1.5, 2.5, -2.5]$
- x_2 :
Среднее: $(1+3+4+6+7-10)/6 = 1.833$
Центрированные значения:
 $[-0.833, 1.167, 2.167, 4.167, 5.167, -11.833]$
- **Центрирование y :**
Среднее: $(-11-2+7+16+26-9)/6 = 4.5$
Центрированные значения:
 $[-15.5, -6.5, 2.5, 11.5, 21.5, -13.5]$
- **Матричное уравнение:**
Матрица X :

1	-1.5	-0.833
1	-1.5	1.167
1	0.5	2.167
1	1.5	4.167
1	2.5	5.167
1	-2.5	-11.833

Вектор Y :

$[-15.5, -6.5, 2.5, 11.5, 21.5, -13.5]$

- **Оценка коэффициентов регрессии:**
Используя метод наименьших квадратов:

$$\beta = (X^T X)^{-1} X^T Y$$

Результат:

$\beta_0, \beta_1, \beta_2$ (конкретные значения рассчитываются в коде).

- **Проверка адекватности модели:**
- Рассчитывается статистика Фишера F.
- Если $F > F_{\text{крит}}$, модель адекватна.
- **Селекция факторов:**
- Проверяется значимость коэффициентов с помощью t-критерия Стьюдента.
- Если p-значение > 0.05 , фактор исключается.
- **Повторная проверка адекватности:**
После исключения незначимых факторов модель проверяется снова.

Результаты работы

В ходе выполнения данной лабораторной работы была написана программа на языке Python 3.12, решающая задачу в общем виде.

```
G:\PROJECTS\GUAP\Programming-GUAP\Processing of experimental data\5>python main.py
Коэффициенты регрессии: [ 4.4408921e-16  1.0953125e+01 -1.1718750e+00]
Статистика F: 4647.954545454542, Критическое значение F: 9.552094495921155
Модель адекватна.
t-статистики: [ 3.21355043e-15  6.58082620e+01 -2.32504713e+01]
p-значения: [1.00000000e+00  7.73158587e-06  1.74297371e-04]
Значимые факторы: [False True True]
Статистика F для значимой модели: 12394.545454545436, Критическое значение F: 7.708647422176786
Модель адекватна.
Коэффициенты значимой модели: [10.953125 -1.171875]
```

Листинг

```
import numpy as np
import scipy.stats as stats

# Данные
x1 = np.array([-2, -1, 0, 1, 2, -3])
x2 = np.array([1, 3, 4, 6, 7, -10])
y = np.array([-11, -2, 7, 16, 26, -9])
```

```

# Центрирование
x1_centered = x1 - np.mean(x1)
x2_centered = x2 - np.mean(x2)
y_centered = y - np.mean(y)

# Шаг 2: Составление матричного уравнения
X = np.column_stack((np.ones(len(x1_centered)), x1_centered, x2_centered)) # Добавляем столбец единиц
Y = y_centered # Используем центрированное y

# Шаг 3: Оценка коэффициентов регрессии
beta = np.linalg.inv(X.T @ X) @ X.T @ Y
print("Коэффициенты регрессии:", beta)

# Шаг 4: Проверка адекватности уравнения регрессии
# Предсказанные значения
Y_pred = X @ beta

# Остатки
residuals = Y - Y_pred

# Средняя квадратичная ошибка (MSE)
MSE = np.mean(residuals**2)

# Общая сумма квадратов (SST)
SST = np.sum((Y - np.mean(Y))**2)

# Сумма квадратов регрессии (SSR)
SSR = np.sum((Y_pred - np.mean(Y))**2)

# Сумма квадратов остатков (SSE)
SSE = np.sum(residuals**2)

# Степени свободы

```

```

n = len(Y)
p = X.shape[1] # Количество параметров (включая свободный член)
df_regression = p - 1
df_residual = n - p

# Средняя квадратичная регрессии (MSR) и средняя квадратичная остатков (MSE)
MSR = SSR / df_regression
MSE = SSE / df_residual

# Статистика F
F_statistic = MSR / MSE
F_critical = stats.f.ppf(1 - 0.05, df_regression, df_residual)

print(f"Статистика F: {F_statistic}, Критическое значение F: {F_critical}")

if F_statistic > F_critical:
    print("Модель адекватна.")
else:
    print("Модель неадекватна.")

# Шаг 5: Селекция факторов по критерию Стьюдента
# Стандартная ошибка коэффициентов
standard_errors = np.sqrt(np.diag(MSE * np.linalg.inv(X.T @ X)))

# t-статистика для каждого коэффициента
t_statistics = beta / standard_errors

# p-значения
p_values = 2 * (1 - stats.t.cdf(np.abs(t_statistics), df_residual))

print("t-статистики:", t_statistics)
print("p-значения:", p_values)

# Уровень значимости

```

```

alpha = 0.05

significant_factors = p_values < alpha

print("Значимые факторы:", significant_factors)

# Шаг 6: Повторная проверка адекватности после исключения незначимых факторов
# Исключаем незначимые факторы
X_significant = X[:, significant_factors]

beta_significant = np.linalg.inv(X_significant.T @ X_significant) @ X_significant.T @ y_centered

# Предсказанные значения для значимой модели
Y_pred_significant = X_significant @ beta_significant
residuals_significant = y_centered - Y_pred_significant

# Остатки
SSE_significant = np.sum(residuals_significant**2)
SSR_significant = np.sum((Y_pred_significant - np.mean(y_centered))**2)

# Степени свободы
n_significant = len(y_centered)
p_significant = X_significant.shape[1]
df_regression_significant = p_significant - 1
df_residual_significant = n_significant - p_significant

# Средняя квадратичная регрессии (MSR) и средняя квадратичная остатков (MSE)
MSR_significant = SSR_significant / df_regression_significant
MSE_significant = SSE_significant / df_residual_significant

# Статистика F для значимой модели
F_statistic_significant = MSR_significant / MSE_significant
F_critical_significant = stats.f.ppf(1 - 0.05, df_regression_significant, df_residual_significant)

print(f"Статистика F для значимой модели: {F_statistic_significant}, Критическое значение F: {F_critical_significant}")

if F_statistic_significant > F_critical_significant:

```

```
print("Модель адекватна.")  
else:  
    print("Модель неадекватна.")  
  
# Выводим коэффициенты значимой модели  
print("Коэффициенты значимой модели:", beta_significant)
```

Выводы

В процессе проведения лабораторной работы гипотеза о неадекватности модели была отклонена, что подтвердило её адекватность. Проверка значимости факторов по критерию Стьюдента выявила незначимость первого фактора. После его исключения модель сохранила адекватность, и гипотеза о неадекватности вновь была отклонена.