ГУАП

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 | | | | | |
| МНОГОФАКТОРНЫЙ РЕГРЕССИОННЫЙ АНАЛИЗ | | | | | |
| по дисциплине: ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ | | | | | |
|  | | | | | |
| РАБОТУ ВЫПОЛНИЛ | | | | | |
| СТУДЕНТ ГР. | 4931 |  |  |  | А.А. Кинько |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Текст задания**

Согласно варианту №4931-12:

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 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 варианта №4931-12*

| **Массив экспериментальных данных** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | -2 | -1 | 0 | 1 | 2 | -0,5 |
|  | 1 | 2 | 3 | 5 | 6 | 7 |
|  | -20 | -11 | -3 | 8 | 14 | -3 |

**Расчетные формулы**

1. Поиск функции регрессии.

В рассматриваемой задаче функция регрессии представляет собой линейный алгебраический полином от двух независимых переменных:

Составляем таблицу экспериментальных данных с центрированными значениями факторов, вычислив средние значения факторов:

Производим центрирование факторов, используя формулу :

*Таблица №2. Расчетная таблица для центрирования факторов*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | -1.917 | -0.917 | 0.083 | 1.083 | 2.083 | -0,417 |
|  | -3 | -2 | 1 | 1 | 2 | 3 |
|  | -20 | -11 | -3 | 8 | 14 | -3 |

Таким образом, функция регрессии принимает вид:

Система нормальных уравнений представляется следующим образом:

Представим и решим систему выше в матричной форме. Матричное уравнение, эквивалентное данной системе, принимает вид:

где

Выражение для вычисления оценок коэффициентов регрессии представляется равенством:

Вычисляем матрицу

Обратная матрица

Далее находим матрицу в правой части уравнения:

Вычисляем оценку вектора коэффициентов регрессии:

Таким образом, получаем следующее уравнение регрессии:

2. Проверка адекватности уравнения

Вычисляем оценки дисперсий, для чего составим расчетную таблицу:

*Таблица №3. Расчетная таблица для проверки адекватности*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| -1.917 | -3 | -20 | -17.5 | 306.25 | -19.65 | -0.35 | 0.123 |
| -0.917 | -2 | -11 | 8.5 | 72.25 | -11.22 | 0.22 | 0.048 |
| 0.083 | 1 | -3 | -0.5 | 0.25 | -2.789 | 0.211 | 0.044 |
| 1.083 | 1 | 8 | 10.5 | 110.25 | 6.554 | 1.446 | 2.091 |
| 2.083 | 2 | 14 | 16.5 | 275.25 | 14.984 | -0.984 | 0.969 |
| -0,417 | 3 | -3 | -0.5 | 0.25 | -2.898 | -0.102 | 0.01 |
|  | | | | |  | | |

Вычисляем оценки средних квадратичных отклонений:

Наблюдаемое значение показателя согласованности:

Для отыскания критического значения показателя согласованности при уровне значимости и степенях свободы , , используя приложение №5, получаем

Поскольку неравенство выполняется, нулевая гипотеза об адекватности функции регрессии экспериментальным данным принимается.

3. Селекция факторов.

Для селекции найдем элементы главной диагонали корреляционной матрицы и умножим каждый из элементов на , получив:

Оценки средних квадратических отклонений коэффициентов принимают значения:

Для каждого фактора находим наблюдаемое значение показателя согласованности:

Для числа степеней свободы и уровня значимости критическое значение показателя согласованности .

Следовательно, , ,

Для данного варианта все факторы значимы, значит пересчет не нужен. В любом другом случае в отношении факторов принимается конкурирующая гипотеза о их незначимости. Тогда в правой части выражения регрессии соответствующие слагаемые необходимо приравнять к нулю и пересчитать это выражение.

**Результат работы**

В ходе выполнения данной лабораторной работы была написана программа на языке Python 3.10, решающая задачу в общем виде для всех вариантов – находит факторы регрессии уравнения в виде линейного алгебраического полинома от двух переменных, проверяет их адекватность и значимость, а также производит перерасчет. Так, для варианта №4931-12 были получены следующие результаты:

Загружаем данные для варианта № 12 ...

Массив x1: [-2.0, -1.0, 0.0, 1.0, 2.0, -0.5]

Массив x2: [1.0, 2.0, 3.0, 5.0, 6.0, 7.0]

Массив y: [-20.0, -11.0, -3.0, 8.0, 14.0, -3.0]

Найдем функцию регрессии в виде алгебраического полинома: y = b0 + b1 \* x1 + b2 \* x2

Среднее значение фактора x1: -0.083

Среднее значение фактора x2: 4.0

Производим центрирование данных:

Массив x1\*: [-1.917, -0.917, 0.083, 1.083, 2.083, -0.417]

Массив x2\*: [-3.0, -2.0, -1.0, 1.0, 2.0, 3.0]

Массив y: [-20.0, -11.0, -3.0, 8.0, 14.0, -3.0]

Для вычисления оценок коэффициентов регрессии решим уравнение B ̃ = (Ẋ^T \* Ẋ)^(-1) (Ẋ^T \* Y)

Матрица (Ẋ^T \* Ẋ):

[[ 6. -0. 0. ]

[-0. 10.21 11.5 ]

[ 0. 11.5 28. ]]

Обратная для этой матрица:

[[ 0.1667 0. 0. ]

[ 0. 0.1823 -0.0749]

[-0. -0.0749 0.0665]]

Матрица (Ẋ^T \* Y):

[[-15. ]

[ 87.255]

[112. ]]

Оценка вектора коэффициентов регрессии:

[[-2.5005 ]

[ 7.5177865]

[ 0.9126005]]

Получаем уравнение: -2.5 + 7.518 ẋ1 + 0.913 ẋ2

---- Проверка адекватности уравнения

Составляем расчетную таблицу:

x1\_i [-1.917, -0.917, 0.083, 1.083, 2.083, -0.417]

x2\_i [-3.0, -2.0, -1.0, 1.0, 2.0, 3.0]

y\_i [-20.0, -11.0, -3.0, 8.0, 14.0, -3.0]

y\_i - y^- [-17.5, -8.5, -0.5, 10.5, 16.5, -0.5]

(y\_i - y^-)^2 [306.25, 72.25, 0.25, 110.25, 272.25, 0.25] - сумма:761.5

y~\_i [-19.65, -11.22, -2.789, 6.554, 14.984, -2.898]

y\_i - y~\_i [-0.35, 0.22, -0.211, 1.446, -0.984, -0.102]

(y\_i - y~\_i)^2 [0.123, 0.048, 0.044, 2.091, 0.969, 0.01] - сумма:3.2849999999999997

Оценка отклонения sigma^2: 152.3

Оценка отклонения sigma1^2: 0.8212499999999999

Значение показателя согласованности: 185.44901065449014

Критическое значение показателя согласованности: 6.26

Так как значение показателя согласованности больше значения показателя F, нулевая гипотеза об адекватности функции регрессии, принимается

---- Селекция факторов

Элементы главной диагонали корреляционной матрицы: [0.13690237499999997, 0.14971387499999997, 0.054613125]

Оценки средних квадратичных отклонений: [0.37000320944553977, 0.3869287725150457, 0.2336945121306874]

Соответствующие показатели согласованности: [6.758049487589768, 19.429380893889636, 3.905100259648602]

Критическое значение показателя согласованности при уровне значимости alpha=0,05 степени свободы f = 3: 3.18

Фактор ẋ0 является значимым

Фактор ẋ1 является значимым

Фактор ẋ2 является значимым

---- Пересчет регрессионного выражения

Все факторы значимы, пересчет не требуется.

Также был получен результат для тестового варианта (64 в файле data.txt):

*Таблица №4. Экспериментальные данные тестового варианта*

| **Массив экспериментальных данных** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | -0.5 | 0 | 0.8 | 0.4 | 0.5 | -0,6 |
|  | -3 | -1 | 2 | 0.5 | 1.5 | 6 |
|  | -15.1 | -1 | -19.9 | 9.5 | 16.5 | 47.9 |

Загружаем данные для варианта № 64 ...

Массив x1: [-0.5, 0.0, 0.8, 0.4, 0.5, 0.6]

Массив x2: [-3.0, -1.0, 2.0, 0.5, 1.5, 6.0]

Массив y: [-15.1, -1.0, 19.9, 9.5, 16.5, 47.9]

Найдем функцию регрессии в виде алгебраического полинома: y = b0 + b1 \* x1 + b2 \* x2

Среднее значение фактора x1: 0.3

Среднее значение фактора x2: 1.0

Производим центрирование данных:

Массив x1\*: [-0.8, -0.3, 0.5, 0.1, 0.2, 0.3]

Массив x2\*: [-4.0, -2.0, 1.0, -0.5, 0.5, 5.0]

Массив y: [-15.1, -1.0, 19.9, 9.5, 16.5, 47.9]

Для вычисления оценок коэффициентов регрессии решим уравнение B ̃ = (Ẋ^T \* Ẋ)^(-1) (Ẋ^T \* Y)

Матрица (Ẋ^T \* Ẋ):

[[ 6. -0. 0. ]

[-0. 1.12 5.85]

[ 0. 5.85 46.5 ]]

Обратная для этой матрица:

[[ 0.1667 0. 0. ]

[ 0. 2.6039 -0.3276]

[-0. -0.3276 0.0627]]

Матрица (Ẋ^T \* Y):

[[ 77.7 ]

[ 40.95]

[325.3 ]]

Оценка вектора коэффициентов регрессии:

[[12.95259 ]

[ 0.061425]

[ 6.98109 ]]

Получаем уравнение: 12.953 + 0.061 ẋ1 + 6.981 ẋ2

---- Проверка адекватности уравнения

Составляем расчетную таблицу:

x1\_i [-0.8, -0.3, 0.5, 0.1, 0.2, 0.3]

x2\_i [-4.0, -2.0, 1.0, -0.5, 0.5, 5.0]

y\_i [-15.1, -1.0, 19.9, 9.5, 16.5, 47.9]

y\_i - y^- [-28.05, -13.95, 6.95, -3.45, 3.55, 34.95]

(y\_i - y^-)^2 [786.802, 194.602, 48.302, 11.902, 12.603, 1221.503] - сумма:2275.714

y~\_i [-15.021, -1.028, 19.964, 9.468, 16.455, 47.876]

y\_i - y~\_i [-0.079, 0.028, -0.064, 0.032, 0.045, 0.024]

(y\_i - y~\_i)^2 [0.006, 0.001, 0.004, 0.001, 0.002, 0.001] - сумма:0.015

Оценка отклонения sigma^2: 455.14279999999997

Оценка отклонения sigma1^2: 0.00375

Значение показателя согласованности: 121371.41333333333

Критическое значение показателя согласованности: 6.26

Так как значение показателя согласованности больше значения показателя F, нулевая гипотеза об адекватности функции регрессии, принимается

---- Селекция факторов

Элементы главной диагонали корреляционной матрицы: [0.0006251249999999999, 0.009764624999999999, 0.000235125]

Оценки средних квадратичных отклонений: [0.025002499875012497, 0.09881611710647205, 0.015333786225195654]

Соответствующие показатели согласованности: [518.0517974102589, 0.6216091240846149, 455.2750310636944]

Критическое значение показателя согласованности при уровне значимости alpha=0,05 степени свободы f = 3: 3.18

Фактор ẋ0 является значимым

Фактор ẋ1 является не значимым

Фактор ẋ2 является значимым

---- Пересчет регрессионного выражения

Пересчет

Найдем функцию регрессии в виде алгебраического полинома: y = b0 + b2 \* x2

Среднее значение фактора x2: 0.0

Производим центрирование данных:

Массив x2\*: [-4.0, -2.0, 1.0, -0.5, 0.5, 5.0]

Массив y: [-15.1, -1.0, 19.9, 9.5, 16.5, 47.9]

Для вычисления оценок коэффициентов регрессии решим уравнение B ̃ = (Ẋ^T \* Ẋ)^(-1) (Ẋ^T \* Y)

Матрица (Ẋ^T \* Ẋ):

[[ 6. 0. ]

[ 0. 46.5]]

Обратная для этой матрица:

[[0.1667 0. ]

[0. 0.0215]]

Матрица (Ẋ^T \* Y):

[[ 77.7]

[325.3]]

Оценка вектора коэффициентов регрессии:

[[12.95259]

[ 6.99395]]

Получаем уравнение: 12.953 + 6.994 ẋ2

---- Проверка адекватности уравнения

Составляем расчетную таблицу:

x1\_i [-0.8, -0.3, 0.5, 0.1, 0.2, 0.3]

x2\_i [-4.0, -2.0, 1.0, -0.5, 0.5, 5.0]

y\_i [-15.1, -1.0, 19.9, 9.5, 16.5, 47.9]

y\_i - y^- [-28.05, -13.95, 6.95, -3.45, 3.55, 34.95]

(y\_i - y^-)^2 [786.802, 194.602, 48.302, 11.902, 12.603, 1221.503] - сумма:2275.714

y~\_i [-15.023, -1.035, 19.947, 9.456, 16.45, 47.922]

y\_i - y~\_i [-0.077, 0.035, -0.047, 0.044, 0.05, -0.022]

(y\_i - y~\_i)^2 [0.006, 0.001, 0.002, 0.002, 0.003, 0.0] - сумма:0.014000000000000002

Оценка отклонения sigma^2: 455.14279999999997

Оценка отклонения sigma1^2: 0.0035000000000000005

Значение показателя согласованности: 130040.79999999997

Критическое значение показателя согласованности: 6.26

Так как значение показателя согласованности больше значения показателя F, нулевая гипотеза об адекватности функции регрессии, принимается

---- Селекция факторов

Элементы главной диагонали корреляционной матрицы: [0.00058345, 7.525e-05]

Оценки средних квадратичных отклонений: [0.0241547096856907, 0.008674675786448736]

Соответствующие показатели согласованности: [536.2345550223334, 806.2491523805069]

Критическое значение показателя согласованности при уровне значимости alpha=0,05 степени свободы f = 3: 3.18

Фактор ẋ0 является значимым

Фактор ẋ2 является значимым

---- Пересчет регрессионного выражения

Все факторы значимы, пересчет не требуется

**Выводы**

В ходе данной лабораторной работы были получены навыки получения уравнения регрессии в виде линейного алгебраического полинома от двух переменных, а также проверка его адекватности и значимости факторов. Так, для варианта №4931-12 было получено уравнение:

**Приложение А. Листинг программы**

Листинг программы доступен по ссылке: <https://github.com/ArtemKinko/SUAI-labs-spring-2023/tree/main/EDP/LABS/LAB-5-Statistic>

Порядок запуска программы описан в файле README.md.

Файл lab5-statistic.py:

import math  
import numpy as np  
  
  
def upload\_matrix(data\_file\_name, task\_file\_name):  
 with open(data\_file\_name) as data\_file:  
 with open(task\_file\_name) as task\_file:  
 task\_num = int(task\_file.readline())  
 print("Загружаем данные для варианта №", task\_num, "...")  
 data = [[], [], []]  
 ready\_data = [[], [], []]  
 for \_ in range(task\_num):  
 data[0] = data\_file.readline().rsplit()  
 data[1] = data\_file.readline().rsplit()  
 data[2] = data\_file.readline().rsplit()  
 ready\_data[0] = [float(x.replace(',', '.')) for x in data[0]]  
 ready\_data[1] = [float(x.replace(',', '.')) for x in data[1]]  
 ready\_data[2] = [float(x.replace(',', '.')) for x in data[2]]  
 return ready\_data  
  
  
def get\_f\_function(f\_function\_file\_name, f1, f2):  
 with open(f\_function\_file\_name) as f\_function\_file:  
 line = ""  
 for \_ in range(f2):  
 line = f\_function\_file.readline().rsplit()  
 return float(line[f1 - 1].replace(",", "."))  
  
  
def get\_t\_function(t\_function\_file\_name, f):  
 with open(t\_function\_file\_name) as t\_function\_file:  
 line = ""  
 for \_ in range(f):  
 line = t\_function\_file.readline().rsplit()  
 return float(line[0].replace(",", "."))  
  
  
def get\_average(x\_matrix):  
 return round(sum(x\_matrix) / len(x\_matrix), 3)  
  
  
def get\_centered\_data(data\_matrix, average\_matrix):  
 return [[round(data\_matrix[0][i] - average\_matrix[0], 3) for i in range(len(data\_matrix[0]))],  
 [round(data\_matrix[1][i] - average\_matrix[1], 3) for i in range(len(data\_matrix[1]))],  
 data\_matrix[2]]  
  
  
def get\_xt\_matrix(data, significance):  
 x\_array = []  
 if significance[0]:  
 x\_array.append([1, 1, 1, 1, 1, 1])  
 if significance[1]:  
 x\_array.append(data[0])  
 if significance[2]:  
 x\_array.append(data[1])  
 return np.matrix(x\_array)  
  
  
def get\_xtx\_matrix(data, significance):  
 x\_matrix = get\_xt\_matrix(data, significance)  
 return x\_matrix.dot(x\_matrix.transpose()).round(2)  
  
  
def get\_inverted\_matrix(matrix):  
 return np.linalg.inv(matrix).round(4)  
  
  
def get\_xty\_matrix(data, significance):  
 x\_matrix = get\_xt\_matrix(data, significance)  
 y\_matrix = np.matrix(data[2]).transpose()  
 return x\_matrix.dot(y\_matrix).round(3)  
  
  
def get\_b\_matrix(xtx\_matrix, xty\_matrix):  
 return xtx\_matrix.dot(xty\_matrix)  
  
  
def get\_y\_from\_equation(x1, x2, coefs, significance):  
 return coefs[0] \* int(significance[0]) + coefs[1] \* int(significance[1]) \* x1 + coefs[2] \* int(significance[2]) \* x2  
  
  
def get\_dispersion\_table(data, coefs, significance):  
 y\_average = get\_average(data[2])  
  
 return [data[0],  
 data[1],  
 data[2],  
 [round(y - y\_average, 3) for y in data[2]],  
 [round((y - y\_average) \*\* 2, 3) for y in data[2]],  
 [round(get\_y\_from\_equation(data[0][i], data[1][i], coefs, significance), 3) for i in  
 range(len(data[0]))],  
 [round(data[2][i] - get\_y\_from\_equation(data[0][i], data[1][i], coefs, significance), 3) for i in  
 range(len(data[0]))],  
 [round((data[2][i] - get\_y\_from\_equation(data[0][i], data[1][i], coefs, significance)) \*\* 2, 3) for i  
 in range(len(data[0]))]]  
  
  
def get\_deviation\_average(sum\_average, denominator):  
 return sum\_average / denominator  
  
  
def get\_deviation\_theo(sum\_theo, denominator):  
 return sum\_theo / denominator  
  
  
def find\_coefs(data, significance):  
 print("\nНайдем функцию регрессии в виде алгебраического полинома: y = " + ("b0" if significance[0] else "")  
 + (" + b1 \* x1" if significance[1] else "") + (" + b2 \* x2" if significance[2] else ""))  
 x1\_average = get\_average(data[0])  
 x2\_average = get\_average(data[1])  
 if significance[1]:  
 print("Среднее значение фактора x1:", x1\_average)  
 if significance[2]:  
 print("Среднее значение фактора x2:", x2\_average)  
 data = get\_centered\_data(data, [x1\_average, x2\_average])  
 print("Производим центрирование данных:")  
 if significance[1]:  
 print("Массив x1\*:", data[0])  
 if significance[2]:  
 print("Массив x2\*:", data[1])  
 print("Массив y:", data[2])  
 print("Для вычисления оценок коэффициентов регрессии решим уравнение B ̃ = (Ẋ^T \* Ẋ)^(-1) (Ẋ^T \* Y)")  
 print("\nМатрица (Ẋ^T \* Ẋ):")  
 xtx\_matrix = get\_xtx\_matrix(data, significance)  
 print(xtx\_matrix)  
 inv\_xtx\_matrix = get\_inverted\_matrix(xtx\_matrix)  
 print("\nОбратная для этой матрица:")  
 print(inv\_xtx\_matrix)  
 print("\nМатрица (Ẋ^T \* Y):")  
 xty\_matrix = get\_xty\_matrix(data, significance)  
 print(xty\_matrix)  
 b\_matrix = get\_b\_matrix(inv\_xtx\_matrix, xty\_matrix)  
 print("\nОценка вектора коэффициентов регрессии:")  
 print(b\_matrix)  
 corrected\_b\_matrix = []  
  
 j = 0  
 titles = ["+", "ẋ1 + ", "ẋ2"]  
 print("\nПолучаем уравнение: ", end="")  
 for i in range(len(significance)):  
 if significance[i]:  
 corrected\_b\_matrix.append(b\_matrix.item(j))  
 print(round(b\_matrix.item(j), 3), titles[i], end=" ")  
 j += 1  
 else:  
 corrected\_b\_matrix.append(0)  
 print("")  
  
 print("\n---- Проверка адекватности уравнения")  
 print("Составляем расчетную таблицу:")  
 dispersion\_table = get\_dispersion\_table(data, corrected\_b\_matrix, significance)  
 titles = ["x1\_i", "x2\_i", "y\_i", "y\_i - y^-", "(y\_i - y^-)^2", "y~\_i", "y\_i - y~\_i", "(y\_i - y~\_i)^2"]  
 sum\_average\_dispersion = sum(dispersion\_table[4])  
 sum\_theo\_dispersion = sum(dispersion\_table[7])  
 for i in range(len(titles)):  
 print(titles[i], dispersion\_table[i], ("- сумма:" + str(sum\_average\_dispersion) if i == 4 else ""),  
 ("- сумма:" + str(sum\_theo\_dispersion) if i == 7 else ""))  
 f2 = 5 - (sum([(int(significance[i])) if i == 0 else 0 for i in range(len(significance))]))  
 deviation = get\_deviation\_average(sum\_average\_dispersion, 5)  
 deviation1 = get\_deviation\_theo(sum\_theo\_dispersion, f2)  
 significance\_value = deviation / deviation1  
 print("Оценка отклонения sigma^2:", deviation)  
 print("Оценка отклонения sigma1^2:", deviation1)  
 print("Значение показателя согласованности:", significance\_value)  
 f\_function = get\_f\_function("f\_function.txt", 5, f2)  
 print("Критическое значение показателя согласованности:", f\_function)  
 if significance\_value > f\_function:  
 print("Так как значение показателя согласованности больше значения показателя F, "  
 "нулевая гипотеза об адекватности функции регрессии, принимается")  
 else:  
 print("Так как значение показателя согласованности меньше значения показателя F, "  
 "нулевая гипотеза об адекватности функции регрессии, принимается")  
  
 print("\n---- Селекция факторов")  
 diag\_matrix = [inv\_xtx\_matrix.item(i, i) for i in range(len(inv\_xtx\_matrix))]  
 diag\_matrix = [deviation1 \* x for x in diag\_matrix]  
 print("Элементы главной диагонали корреляционной матрицы:", diag\_matrix)  
 sigmas = [math.sqrt(x) for x in diag\_matrix]  
 print("Оценки средних квадратичных отклонений:", sigmas)  
 t\_function\_a = [abs(b\_matrix[i][0]) / sigmas[i] for i in range(len(sigmas))]  
 print("Соответствующие показатели согласованности:", t\_function\_a)  
 t\_function = get\_t\_function("t\_function.txt", 3)  
 print("Критическое значение показателя согласованности при уровне значимости alpha=0,05 степени свободы f = 3:",  
 t\_function)  
  
 new\_significance = []  
 titles = ["ẋ0", "ẋ1", "ẋ2"]  
 j = 0  
 for i in range(3):  
 if not significance[i]:  
 new\_significance.append(False)  
 else:  
 new\_significance.append(t\_function\_a[j] > t\_function)  
 if t\_function\_a[j] > t\_function:  
 print("Фактор", titles[i], "является значимым")  
 else:  
 print("Фактор", titles[i], "является не значимым")  
 j += 1  
  
 print("\n---- Пересчет регрессионного выражения")  
 if new\_significance.count(False) == 0 or new\_significance == significance:  
 print("Все факторы значимы, пересчет не требуется")  
 return  
 else:  
 if new\_significance.count(True) == 0:  
 print("Все коэффициенты не значимы. Пересчет невозможен")  
 return  
 print("Пересчет")  
 find\_coefs(data, new\_significance)  
  
  
def solve\_task():  
 data = upload\_matrix("data.txt", "task.txt")  
 print("Массив x1:", data[0])  
 print("Массив x2:", data[1])  
 print("Массив y:", data[2])  
 significance = [True, True, True]  
 find\_coefs(data, significance)  
  
  
solve\_task()

Файл data.txt:

2,1 2,8 3,5 4 5 5,5  
5 4 3,1 3 2 1  
2,5 6,5 10 11,8 17 19,5  
5 6 7 4 2 1  
1,5 1,1 0,5 1 0 -1  
26,7 32 37 22,2 13 8  
4 2 0 1 3 -1  
2,5 3 4 1 1,2 2  
6,2 5 4,8 0,2 2 0  
0,5 1 2 4 3,5 -1  
0 1 1,5 2 2,5 1  
0,2 5 8,8 15,2 17 1  
1 2 3 4 4,5 -2  
-1 0 1,5 2 3 -2  
9,2 10,7 12 15,2 14,6 -1  
2 1,5 0 1 3 4  
-1 0,5 0,1 0 1 3  
6,2 1 0,6 2,1 1 -3,5  
1 0,5 3 2 1 -2  
1 2 2 3 0,3 0,5  
2 4,3 7,2 8 0 -3  
2 1 0,5 2 3 -0,5  
-1 0 1 2 1,8 0,5  
9,7 5,1 1,5 4,2 6,8 -0,5  
1 0,5 0 2 1,5 3  
1 0,5 1 3 2 4  
4,2 3,4 0 2,1 3,2 2,5  
2 3 4 6 7 -1  
-2 0 1 2 3 -1  
16 6 1 -2 -8 7  
-3 2 0 1 2 -2  
0 1 2 3 4 5  
-17 -1 -4,5 2 8,3 -1  
-2 -1 0 1 2 -0,5  
1 2 3 5 6 7  
-20 -11 -3 8 14 -3  
-3 -1 0 2 5 6  
-2 3 5 7 9 8  
-15 -9 -7 -2 2 4  
-2 -1 0 1 2 -3  
1 3 4 6 7 -10  
-11 -2 7 16 26 -9  
-2 -1 0 1 2 1,5  
3 2 5 6 7 1  
-2 2,5 0 2 2 11  
-2 -1 0 1 2 0,5  
4 0 5 2 0 1  
-6,8 1 1 5 12 5,7  
-3 -2 1,5 2 3 4  
4 1 2 -1 3 -2  
-17 -7 11,6 18,5 20 30  
-2 -1,5 -1 1 2 3  
2 1 0 3 1,5 4  
4 -2,5 -9 14 4,5 23  
-2 -1 0 1 2 4  
2,5 3 -4 -2 1 -1  
-22 -23 30 17 -5 13  
2 1,5 1 3 -1 4  
-1 2 0,5 -3 4 -2  
17 9 8 25 -12 32  
2 0,5 1 -1 4 3  
-3 -2 -1 2 1 -0,5  
-17 -5,5 -8 13 -22 -21  
1 2 -3 4 3 1,5  
-3 -2 2 -1 1 0  
0 3 -6 9 11 4  
2 3 0,5 -1 -2 1  
4 -1 1 2 3 -4  
0 13 1 -9 -12 11  
-2 1,5 3 -1 2 4  
3 -1 -2 -1 -3 -0,5  
17 -6 -13 -8 -21 0  
-4 -1 0 2 1 -2  
-3 -2 2 3 4 -1  
-16 -3 -3 5 -2 -9  
4 2 0 1 3 -1  
2,5 3 4 1 1,2 2  
6,2 5 4,8 0,2 2 0  
0,5 1 2 4 3,5 -1  
0 1 1,5 2 2,5 1  
0,2 5 8,8 15,2 17 1  
1 2 3 4 4,5 -2  
-1 0 1,5 2 3 -2  
9,2 10,7 12 15,2 14,6 -1  
2 1,5 0 1 3 4  
-1 0,5 0,1 0 1 3  
6,2 1 0,6 2,1 1 -3,5  
1 0,5 3 2 1 -2  
1 2 2 3 0,3 0,5  
2 4,3 7,2 8 0 -3  
2 1 0,5 2 3 -0,5  
-1 0 1 2 1,8 0,5  
9,7 5,1 1,5 4,2 6,8 -0,5  
1 0,5 0 2 1,5 3  
1 0,5 1 3 2 4  
4,2 3,4 0 2,1 3,2 2,5  
2 3 4 6 7 -1  
-2 0 1 2 3 -1  
16 6 1 -2 -8 7  
0,5 1 2 4 3,5 -1  
0 1 1,5 2 2,5 1  
0,2 5 8,8 15,2 17 1  
-2 -1 0 1 2 -0,5  
1 2 3 5 6 7  
-20 -11 -3 8 14 -3  
-2 -1 0 1 2 -3  
1 3 4 6 7 -10  
-11 -2 7 16 26 -9  
-2 -1 0 1 2 1,5  
3 2 5 6 7 1  
-2 2,5 0 2 2 11  
-2 -1 0 1 2 0,5  
4 0 5 2 0 1  
-6,8 1 1 5 12 5,7  
-3 -2 1,5 2 3 4  
4 1 2 -1 3 -2  
-17 -7 11,6 18,5 20 30  
-2 -1 0 1 2 4  
2,5 3 -4 -2 1 -1  
-22 -23 30 17 -5 13  
2 1,5 1 3 -1 4  
-1 2 0,5 -3 4 -2  
17 9 8 25 -12 32  
2 0,5 1 -1 4 3  
-3 -2 -1 2 1 -0,5  
-17 -5,5 -8 13 -22 -21  
1 2 -3 4 3 1,5  
-3 -2 2 -1 1 0  
0 3 -6 9 11 4  
2 3 0,5 -1 -2 1  
4 -1 1 2 3 -4  
0 13 1 -9 -12 11  
-2 1,5 3 -1 2 4  
3 -1 -2 -1 -3 -0,5  
17 -6 -13 -8 -21 0  
-4 -1 0 2 1 -2  
-3 -2 2 3 4 -1  
-16 -3 -3 5 0 -9  
2 3 4 6 7 -1  
-2 0 1 2 3 -1  
16 6 1 -2 -8 7  
-3 2 0 1 2 -2  
0 1 2 3 4 5  
-17 -1 -4,5 2 8,3 -1  
-2 -1 0 1 2 4  
2,5 3 -4 -2 1 -1  
-22 -23 30 17 -5 13  
2 1,5 1 3 -1 4  
-1 2 0,5 -3 4 -2  
17 9 8 25 -12 32  
2 0,5 1 -1 4 3  
-3 -2 -1 2 1 -0,5  
-17 -5,5 -8 13 -22 -21  
1 -1 0 -2 2 3  
-2 0 1 2 3 4  
10 -3 -2 -2 -12 -22  
1 2 -3 4 3 1,5  
-3 -2 2 -1 1 0  
0 3 -6 9 11 4  
2 3 0,5 -1 -2 1  
4 -1 1 2 3 -4  
0 13 1 -9 -12 11  
-2 1,5 3 -1 2 4  
3 -1 -2 -1 -3 -0,5  
17 -6 -13 -8 -21 0  
-4 -1 0 2 1 -2  
-3 -2 2 3 4 -1  
-16 -3 -3 5 0 -9  
2 3 4 6 7 -1  
-2 0 1 2 3 -1  
16 6 1 -2 -8 7  
-3 2 0 1 2 -2  
0 1 2 3 4 5  
-17 -1 -4,5 2 8,3 -1  
1 2 3 4 4,5 -2  
-1 0 1,5 2 3 -2  
9,2 10,7 12 15,2 14,6 -1  
2 3 0,5 -1 -2 1  
4 -1 1 2 3 -4  
0 13 1 -9 -12 11  
-2 1,5 3 -1 2 4  
3 -1 -2 -1 -3 -0,5  
17 -6 -13 -8 -21 0  
-4 -1 0 2 1 -2  
-3 -2 2 3 4 -1  
-16 -3 -3 5 -2 -9  
4 2 0 1 3 -1  
2,5 3 4 1 1,2 2  
6,2 5 4,8 0,2 2 0  
-0,5 0 0,8 0,4 0,5 0,6  
-3 -1 2 0,5 1,5 6  
-15,1 -1 19,9 9,5 16,5 47,9

Файл f\_function.txt:

161 200 216 225 230 234 237 239 241 242 243 244  
18,51 19 19,16 19,25 19,3 19,33 19,36 19,37 19,38 19,39 19,4 19,41  
10,13 9,55 9,28 9,12 9,01 8,94 8,88 8,84 8,81 8,78 8,76 8,74  
7,71 6,94 6,59 6,39 6,26 6,16 6,09 6,04 6 5,96 5,93 5,91  
6,61 5,79 5,41 5,19 5,05 4,95 4,88 4,82 4,78 4,74 4,7 4,68  
5,99 5,14 4,76 4,53 4,39 4,28 4,21 4,15 4,1 4,06 4,03 4  
5,59 4,74 4,35 4,12 3,97 3,87 3,79 3,73 3,68 3,63 3,6 3,57  
5,32 4,46 4,07 3,84 3,69 3,58 3,5 3,44 3,39 3,34 3,31 3,28  
5,12 4,26 3,86 3,63 3,48 3,37 3,29 3,23 3,18 3,13 3,1 3,07  
4,96 4,1 3,71 3,48 3,33 3,22 3,14 3,07 3,02 2,97 2,94 2,91  
4,84 3,98 3,59 3,36 3,2 3,09 3,01 2,95 2,9 2,86 2,82 2,79  
4,75 3,88 3,49 3,26 3,11 3 2,92 2,85 2,8 2,76 2,72 2,69  
4,67 3,8 3,41 3,18 3,02 2,92 2,84 2,77 2,72 2,67 2,63 2,6  
4,6 3,74 3,34 3,11 2,96 2,85 2,77 2,7 2,65 2,6 2,56 2,53  
4,54 3,68 3,29 3,06 2,9 2,79 2,7 2,64 2,59 2,55 2,51 2,48  
4,49 3,63 3,24 3,01 2,85 2,74 2,66 2,59 2,54 2,49 2,45 2,42  
4,45 3,59 3,2 2,96 2,81 2,7 2,62 2,55 2,5 2,45 2,41 2,38  
3,84 3 2,6 2,37 2,21 2,1 2,01 1,94 1,88 1,83 1,79 1,75

Файл t\_function.txt:

12,7  
4,3  
3,18  
2,78  
2,57  
2,45  
2,36  
2,31  
2,26  
2,23  
2,2  
2,18  
2,16  
2,14  
2,13  
2,12  
2,11  
2,1  
2,09  
2,09  
2,08  
2,07  
2,07  
2,06  
2,06  
2,06  
2,05  
2,05  
2,05  
2,04

Файл task.txt:

12