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 №4 | | | | | |
| однофакторный 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 *алгебраического полинома второй степени*;
* проверить адекватность уравнения регрессии;
* проверить значимость коэффициентов регрессии.

Расчеты произвести в скалярной и матричной форме.

Порядок выполнения задания:

1. Составить систему нормальных уравнений, используя массив экспериментальных данных (таблица 1);
2. Найти оценки коэффициентов регрессии посредством решения системы нормальных уравнений;
3. При расчетах в матричной форме составить матричное уравнение с вектором неизвестных оценок коэффициентов регрессии и найти его решение;
4. Проверить адекватность построенного уравнения регрессии экспериментальным данным по критерию Фишера при уровне значимости α = 0,01;
5. Проверить значимость коэффициентов регрессии по критерию Стьюдента при таком же уровне значимости;
6. Повторно проверить адекватность уравнения регрессии после исключения незначимых коэффициентов.

*Таблица №1. Экспериментальные данные варианта №4931-12*

| **Массив экспериментальных данных** | | | | | |
| --- | --- | --- | --- | --- | --- |
| x | -2 | 0 | 1 | 2 | 4 |
| y | 18 | 3 | 1 | 3 | 21 |

**Расчетные формулы**

1. Поиск регрессионных коэффициентов (скалярная форма).

Класс функций задан в условии задания – полином второй степени

где [1]

Для заданного количества опытов (5) система нормальных уравнений принимает вид:

С учетом [1] система уравнений [2] преобразуется следующим образом:

Оценки коэффициентов уравнения регрессии находятся по формулам:

*Таблица №2. Расчетная таблица скалярного вычисления коэффициентов уравнения*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| -2 | 18 | 4 | -8 | 16 | -36 | 72 |
| 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 4 | 8 | 16 | 6 | 12 |
| 4 | 21 | 16 | 64 | 256 | 84 | 336 |
|  |  |  |  |  |  |  |

Вычисляем оценки коэффициентов:

2. Проверка адекватности регрессионной зависимости экспериментальным данным.

Для этого необходимо вычислить оценки дисперсий.

*Таблица №3. Расчетная таблица вычисления оценок дисперсий*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| -2 | 18 | 8.8 | 77.44 | 18.145 | -0.145 | 0.021 |
| 0 | 3 | -6.2 | 38.44 | 2.571 | 0.429 | 0.184 |
| 1 | 1 | 8.2 | 67.24 | 0.961 | 0.039 | 0.002 |
| 2 | 3 | -6.2 | 38.44 | 3.469 | -0.469 | 0.22 |
| 4 | 21 | 11.8 | 139.24 | 20.839 | 0.161 | 0.026 |
|  | | | |  | | |

Оценки дисперсий вычисляются по формулам:

Наблюдаемое значение показателя согласованности:

Критическое значение показателя согласованности при уровне значимости и степенях свободы находим в приложении 5:

Так как , нулевая гипотеза о соответствии функции регрессии экспериментальным данных принимается.

3. Проверка значимости коэффициентов уравнения регрессии.

Для вычисления наблюдаемых значений показателя согласованности необходимо найти диагональные элементы корреляционный матрицы:

Находим произведение транспонированной матрицы на исходную:

Далее требуется найти элементы главной диагонали матрицы . Главная диагональ корреляционной матрицы вектора оценок коэффициентов регрессии:

Вычисляем оценки средних квадратических отклонений коэффициентов

наблюдаемые значения показателя:

Критическое значение показателя согласованности при уровне значимости и степени свободы находим в приложении 6:

Проверяем условия и получаем:

На основе приведенных неравенств делаем вывод, что коэффициенты являются значимыми, а коэффициент принимается равным нулю.

Окончательный вид уравнения регрессии:

Повторная проверка адекватности по критерию Фишера проводится аналогично пункту №2.

4. Поиск регрессионных коэффициентов (матричная форма)

Матричное уравнение имеет вид:

где матрицы и вычислены в пункте №2,

Выражение для вычисления оценок коэффициентов регрессии представляется в виде:

Вычислим правую часть уравнения:

Найдем оценки коэффициентов регрессии: 0.01190476 -0.02380952 -0.03571429]

[-0.02380952 0.09761905 0.02142857]

[-0.03571429 0.02142857 0.35714286]]

Так же, как и в пункте №1 при расчете в скалярной форме, получилось уравнение регрессии:

**Результат выполнения.**

В ходе выполнения данной лабораторной работы была написана программа на языке Python 3.10, решающая задачу в общем виде для всех вариантов – находит коэффициенты регрессии уравнения в виде алгебраического полинома второй степени, проверяет их адекватность и значимость. Так, для варианта №4931-12 были получены следующие результаты:

Загружаем данные для варианта № 12 ...

Значения X: [-2.0, 0.0, 1.0, 2.0, 4.0]

Значения Y: [18.0, 3.0, 1.0, 3.0, 21.0]

Необходимо построить уравнение регрессии y = f(x) в предположении,что оно является алгебраическим полиномом второй степени

Пусть y = a0 \* f0(x) + a1 \* f1(x) + a2 \* f2(x), где

f0(x) = x^2, f1(x) = x, f2(x) = 1

---- Расчетная таблица для скалярного расчета:

x\_i [-2.0, 0.0, 1.0, 2.0, 4.0] - сумма: 5.0

y\_i [18.0, 3.0, 1.0, 3.0, 21.0] - сумма: 46.0

x^2\_i [4.0, 0.0, 1.0, 4.0, 16.0] - сумма: 25.0

x^3\_i [-8.0, 0.0, 1.0, 8.0, 64.0] - сумма: 65.0

x^4\_i [16.0, 0.0, 1.0, 16.0, 256.0] - сумма: 289.0

y\_i \* x\_i [-36.0, 0.0, 1.0, 6.0, 84.0] - сумма: 55.0

y\_i \* x^2\_i [72.0, 0.0, 1.0, 12.0, 336.0] - сумма: 421.0

Матрица A:

[[289. 65. 25.]

[ 65. 25. 5.]

[ 25. 5. 5.]]

Определитель |A|: 8400

Матрица A0:

[[421. 65. 25.]

[ 55. 25. 5.]

[ 46. 5. 5.]]

Определитель |A0|: 17299

Коэффициент a0: 2.059

Матрица A1:

[[289. 421. 25.]

[ 65. 55. 5.]

[ 25. 46. 5.]]

Определитель |A1|: -30820

Коэффициент a1: -3.669

Матрица A2:

[[289. 65. 421.]

[ 65. 25. 55.]

[ 25. 5. 46.]]

Определитель |A2|: 21600

Коэффициент a2: 2.571

[2.059, -3.669, 2.571]

Функция, полученная с помощью скалярного расчета, принимает вид:

2.059 \* x^2 + -3.669 \* x + 2.571

---- Произведем расчеты в матричной форме

Решим уравнение A<3> = (F^T[3;5] F[5;3])^-1 F^T[3;5] Y<5>

Вычисляем обратную матрицу для (F^T F)^-1 для матрицы:

[[289. 65. 25.]

[ 65. 25. 5.]

[ 25. 5. 5.]]

Обратная матрица:

[[ 0.01190476 -0.02380952 -0.03571429]

[-0.02380952 0.09761905 0.02142857]

[-0.03571429 0.02142857 0.35714286]]

Вычисляем матрицу в правой части уравнения - F^T\_[5;3] Y<5>:

F^T\_[5;3]:

[[ 4. 0. 1. 4. 16.]

[-2. 0. 1. 2. 4.]

[ 1. 1. 1. 1. 1.]]

Y<5>:

[[18.]

[ 3.]

[ 1.]

[ 3.]

[21.]]

Полученная правая часть уравнения:

[[421.]

[ 55.]

[ 46.]]

Получим оценки коэффициенты регрессии путем перемножения обратной матрицы с полученной:

Функция, полученная с помощью скалярного расчета, принимает вид:

2.06 \* x^2 + -3.669 \* x + 2.571

---- Проверяем адекватность регрессионной зависимости экспериментальным данным

Расчетная таблица:

x\_i [-2.0, 0.0, 1.0, 2.0, 4.0]

y\_i [18.0, 3.0, 1.0, 3.0, 21.0]

y\_i - y^\* [8.8, -6.2, -8.2, -6.2, 11.8]

(y\_i - y^\*)^2 [77.44, 38.44, 67.24, 38.44, 139.24]

Сумма квадратов разности значения y и оценки мат. ожидания - 360.8

y~\_i [18.145, 2.571, 0.961, 3.469, 20.839]

y\_i - y~\_i [-0.145, 0.429, 0.039, -0.469, 0.161]

(y\_i - y~\_i)^2 [0.021, 0.184, 0.002, 0.22, 0.026]

Сумма квадратов разности значения y и значений полученной функции - 0.453

Вычисляем оценку дисперсий sigma^2: 90.2

Вычисляем оценку дисперсий sigma^2\_1: 0.2265

Значение показателя согласованности: 398.233995584989

Критическое значение показателя согласованности при уровне значимости alpha=0.01 и степенях свободы: f1 = 4,f2 = 2: 99.25

Так как показатель согласованности больше критической точки распределения Фишера,нулевая гипотеза H0 о соответствии функции регрессии экспериментальным данным принимается

---- Проверяем значимость коэффициентов регрессии.

Элементы главной диагонали обратной матрицы: [0.011904761904761906, 0.09761904761904763, 0.35714285714285715]

Перемножим sigma^2\_1 на элементы диагонали и получим дисперсии коэффициентов регрессии:

[0.002696428571428572, 0.02211071428571429, 0.08089285714285714]

Соответствующие средние квадратичные отклонения: [0.05192714676764526, 0.1486967191491268, 0.28441669631520783]

Находим наблюдаемые значения показателя: [39.651706827129594, 24.674384350877226, 9.039553701695002]

Критическое значение показателя согласованности при уровне значимости alpha=0,01 степени свободы f = 2: 9.92

Сравниваем полученные показатели с критическим значением:

Коэффициент a0 является значимым

Коэффициент a1 является значимым

Коэффициент a2 является незначимым

Исключая, незначимые коэффициенты получим уравнение:

2.059 \* x^2 + -3.669 \* x

---- Повторно проверим адекватность для нового уравнения

---- Проверяем адекватность регрессионной зависимости экспериментальным данным

Расчетная таблица:

x\_i [-2.0, 0.0, 1.0, 2.0, 4.0]

y\_i [18.0, 3.0, 1.0, 3.0, 21.0]

y\_i - y^\* [8.8, -6.2, -8.2, -6.2, 11.8]

(y\_i - y^\*)^2 [77.44, 38.44, 67.24, 38.44, 139.24]

Сумма квадратов разности значения y и оценки мат. ожидания - 360.8

y~\_i [15.574, 0.0, -1.61, 0.898, 18.268]

y\_i - y~\_i [2.426, 3.0, 2.61, 2.102, 2.732]

(y\_i - y~\_i)^2 [5.885, 9.0, 6.812, 4.418, 7.464]

Сумма квадратов разности значения y и значений полученной функции - 33.579

Вычисляем оценку дисперсий sigma^2: 90.2

Вычисляем оценку дисперсий sigma^2\_1: 16.7895

Значение показателя согласованности: 5.372405372405373

Критическое значение показателя согласованности при уровне значимости alpha=0.01 и степенях свободы: f1 = 4,f2 = 2: 99.25

Так как показатель согласованности меньше критической точки распределения Фишера,нулевая гипотеза H0 о соответствии функции регрессии экспериментальным данным отклоняется

**Выводы**

В ходе данной лабораторной работы были получены навыки получения уравнения регрессии в виде полинома второй степени, а также проверки его адекватности и значимости его коэффициентов. 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-4-Statistic>

Порядок запуска программы описан в файле README.md.

Файл lab4-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()  
 ready\_data[0] = [float(x.replace(',', '.')) for x in data[0]]  
 ready\_data[1] = [float(x.replace(',', '.')) for x in data[1]]  
 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\_needed\_matrix(raw\_data):  
 ready\_data = [raw\_data[0], raw\_data[1], [x \*\* 2 for x in raw\_data[0]], [x \*\* 3 for x in raw\_data[0]],  
 [x \*\* 4 for x in raw\_data[0]], [raw\_data[0][i] \* raw\_data[1][i] for i in range(len(raw\_data[0]))],  
 [raw\_data[0][i] \*\* 2 \* raw\_data[1][i] for i in range(len(raw\_data[0]))]]  
 return ready\_data, [sum(line) for line in ready\_data]  
  
  
def get\_coefs(summaries):  
 A = np.matrix([[summaries[4], summaries[3], summaries[2]],  
 [summaries[3], summaries[2], summaries[0]],  
 [summaries[2], summaries[0], 5]])  
 det\_A = int(np.linalg.det(A))  
 print("\nМатрица A:")  
 print(A)  
 print("Определитель |A|:", det\_A)  
  
 A0 = np.matrix([[summaries[6], summaries[3], summaries[2]],  
 [summaries[5], summaries[2], summaries[0]],  
 [summaries[1], summaries[0], 5]])  
 det\_A0 = int(np.linalg.det(A0))  
 a0 = round(det\_A0 / det\_A, 3)  
 print("\nМатрица A0:")  
 print(A0)  
 print("Определитель |A0|:", det\_A0)  
 print("Коэффициент a0:", a0)  
  
 A1 = np.matrix([[summaries[4], summaries[6], summaries[2]],  
 [summaries[3], summaries[5], summaries[0]],  
 [summaries[2], summaries[1], 5]])  
 det\_A1 = int(np.linalg.det(A1))  
 a1 = round(det\_A1 / det\_A, 3)  
 print("\nМатрица A1:")  
 print(A1)  
 print("Определитель |A1|:", det\_A1)  
 print("Коэффициент a1:", a1)  
  
 A2 = np.matrix([[summaries[4], summaries[3], summaries[6]],  
 [summaries[3], summaries[2], summaries[5]],  
 [summaries[2], summaries[0], summaries[1]]])  
 det\_A2 = int(np.linalg.det(A2))  
 a2 = round(det\_A2 / det\_A, 3)  
 print("\nМатрица A2:")  
 print(A2)  
 print("Определитель |A2|:", det\_A2)  
 print("Коэффициент a2:", a2)  
  
 return [a0, a1, a2], A  
  
  
def get\_table\_dispersion(data, coefs):  
 average\_y = sum(data[1]) / len(data[1])  
 diff\_average = [y - average\_y for y in data[1]]  
 sqr\_diff\_average = [y \*\* 2 for y in diff\_average]  
 sum\_sqr\_diff\_average = sum(sqr\_diff\_average)  
  
 theoretical\_y = [coefs[0] \* x \*\* 2 + coefs[1] \* x + coefs[3] for x in data[0]]  
 diff\_theo = [data[1][i] - theoretical\_y[i] for i in range(len(theoretical\_y))]  
 sqr\_diff\_theo = [y \*\* 2 for y in diff\_theo]  
 sum\_sqr\_diff\_theo = sum(sqr\_diff\_theo)  
  
  
def get\_f\_y\_matrix(data):  
 left = np.matrix([data[2], data[0], [1, 1, 1, 1, 1]])  
 right = np.matrix([[x] for x in data[1]])  
 return left, right  
  
  
def get\_table\_adequacy(data, theo\_coefs, significances):  
 y\_average = sum(data[1]) / len(data[1])  
 theo\_y = [round(theo\_coefs[0] \* significances[0] \* x \*\* 2 +  
 theo\_coefs[1] \* significances[1] \* x +  
 theo\_coefs[2] \* significances[2], 3) for x in data[0]]  
 return [data[0],  
 data[1],  
 [round(y - y\_average, 3) for y in data[1]],  
 [round((y - y\_average) \*\* 2, 3) for y in data[1]],  
 theo\_y,  
 [round(data[1][i] - theo\_y[i], 3) for i in range(len(data[1]))],  
 [round((data[1][i] - theo\_y[i]) \*\* 2, 3) for i in range(len(data[1]))]]  
  
  
def check\_agreement(new\_data, coefs, significance):  
 print("\n\n---- Проверяем адекватность регрессионной зависимости экспериментальным данным")  
 titles\_adequacy = ["x\_i\t\t\t\t", "y\_i\t\t\t\t", "y\_i - y^\*\t\t", "(y\_i - y^\*)^2\t", "y~\_i\t\t\t", "y\_i - y~\_i\t\t",  
 "(y\_i - y~\_i)^2\t"]  
 table\_adequacy = get\_table\_adequacy(new\_data, coefs, significance)  
 table\_adequacy\_sums = [sum(table\_adequacy[3]), sum(table\_adequacy[6])]  
 print("Расчетная таблица:")  
 for i in range(len(table\_adequacy)):  
 print(titles\_adequacy[i], table\_adequacy[i])  
 if i == 3:  
 print("Сумма квадратов разности значения y и оценки мат. ожидания - ", table\_adequacy\_sums[0])  
 if i == 6:  
 print("Сумма квадратов разности значения y и значений полученной функции - ", table\_adequacy\_sums[1])  
 dispersion = table\_adequacy\_sums[0] / 4  
 dispersion\_1 = table\_adequacy\_sums[1] / 2  
 print("Вычисляем оценку дисперсий sigma^2:", dispersion)  
 print("Вычисляем оценку дисперсий sigma^2\_1:", dispersion\_1)  
 agreement\_coef = dispersion / dispersion\_1  
 print("Значение показателя согласованности:", agreement\_coef)  
 f\_function = get\_f\_function("f\_function.txt", 4, 2)  
 print("Критическое значение показателя согласованности при уровне значимости alpha=0.01 и степенях свободы: f1 = 4,"  
 "f2 = 2:", f\_function)  
 if agreement\_coef > f\_function:  
 print("Так как показатель согласованности больше критической точки распределения Фишера,"  
 "нулевая гипотеза H0 о соответствии функции регрессии экспериментальным данным принимается")  
 else:  
 print("Так как показатель согласованности меньше критической точки распределения Фишера,"  
 "нулевая гипотеза H0 о соответствии функции регрессии экспериментальным данным отклоняется")  
 return dispersion\_1  
  
  
def solve\_task():  
 data = upload\_matrix("data.txt", "task.txt")  
 print("Значения X:", data[0])  
 print("Значения Y:", data[1])  
 print("Необходимо построить уравнение регрессии y = f(x) в предположении,"  
 "что оно является алгебраическим полиномом второй степени")  
 print("Пусть y = a0 \* f0(x) + a1 \* f1(x) + a2 \* f2(x), где \n"  
 "f0(x) = x^2, f1(x) = x, f2(x) = 1")  
 new\_data, summary = get\_needed\_matrix(data)  
 print("\n\n---- Расчетная таблица для скалярного расчета:")  
 titles = ["x\_i\t\t\t", "y\_i\t\t\t", "x^2\_i\t\t", "x^3\_i\t\t", "x^4\_i\t\t", "y\_i \* x\_i\t", "y\_i \* x^2\_i\t"]  
 for i in range(len(new\_data)):  
 print(titles[i], new\_data[i], "\t- сумма:", summary[i])  
 coefs, a\_matrix = get\_coefs(summary)  
 print(coefs)  
 print("\nФункция, полученная с помощью скалярного расчета, принимает вид:")  
 print(coefs[0], "\* x^2 +", coefs[1], "\* x +", coefs[2])  
  
 print("\n---- Произведем расчеты в матричной форме")  
 print("Решим уравнение A<3> = (F^T[3;5] F[5;3])^-1 F^T[3;5] Y<5>")  
 print("\nВычисляем обратную матрицу для (F^T F)^-1 для матрицы:")  
 print(a\_matrix)  
 print("\nОбратная матрица:")  
 inv\_a\_matrix = np.linalg.inv(a\_matrix)  
 print(inv\_a\_matrix)  
 print("Вычисляем матрицу в правой части уравнения - F^T\_[5;3] Y<5>:")  
 left\_matrix, right\_matrix = get\_f\_y\_matrix(new\_data)  
 print("\nF^T\_[5;3]:")  
 print(left\_matrix)  
 print("\nY<5>:")  
 print(right\_matrix)  
 print("\nПолученная правая часть уравнения:")  
 total\_matrix = left\_matrix.dot(right\_matrix)  
 print(total\_matrix)  
 print("\nПолучим оценки коэффициенты регрессии путем перемножения обратной матрицы с полученной:")  
 coefs\_matrix = [round(float(x[0]), 3) for x in inv\_a\_matrix.dot(total\_matrix)]  
 print("\nФункция, полученная с помощью скалярного расчета, принимает вид:")  
 print(coefs\_matrix[0], "\* x^2 +", coefs\_matrix[1], "\* x +", coefs\_matrix[2])  
  
 dispersion\_1 = check\_agreement(new\_data, coefs, [1, 1, 1])  
  
 print("\n\n---- Проверяем значимость коэффициентов регрессии.")  
 diag\_matrix = [inv\_a\_matrix.item(0, 0), inv\_a\_matrix.item(1, 1), inv\_a\_matrix.item(2, 2)]  
 print("Элементы главной диагонали обратной матрицы:", diag\_matrix)  
 print("Перемножим sigma^2\_1 на элементы диагонали и получим дисперсии коэффициентов регрессии:")  
 diag\_matrix = [dispersion\_1 \* x for x in diag\_matrix]  
 print(diag\_matrix)  
 sigmas\_matrix = [math.sqrt(x) for x in diag\_matrix]  
 print("Соответствующие средние квадратичные отклонения:", sigmas\_matrix)  
 t\_function\_a = [abs(coefs[i]) / sigmas\_matrix[i] for i in range(len(sigmas\_matrix))]  
 print("Находим наблюдаемые значения показателя:", t\_function\_a)  
 t\_function = get\_t\_function("t\_function.txt", 2)  
 print("Критическое значение показателя согласованности при уровне значимости alpha=0,01 степени свободы f = 2:",  
 t\_function)  
 print("Сравниваем полученные показатели с критическим значением:\n")  
 significance = [x > t\_function for x in t\_function\_a]  
 for i in range(len(significance)):  
 print("Коэффициент a" + str(i) + " является", ("значимым" if significance[i] else "незначимым"))  
 print("Исключая, незначимые коэффициенты получим уравнение:")  
 print((str(coefs[0]) + " \* x^2" if significance[0] == 1 else "") +  
 (" + " + str(coefs[1]) + " \* x" if significance[1] == 1 else "") +  
 (" + " + str(coefs[2]) if significance[2] == 1 else ""))  
  
 print("\n\n---- Повторно проверим адекватность для нового уравнения")  
 check\_agreement(new\_data, coefs, significance)  
  
  
solve\_task()

Файл data.txt:

-2 0 1 2 3  
23 2 0 4 15  
-3 -1 0 1 2  
10 -3 -3 1 11  
-2 0 1 2 3  
-8 -5 1 8 18  
-2 0 2 3 4  
14 1 -3 -2 1  
-3 -2 0 1 2  
-11 -5 3 3 3  
-4 -2 0 1 2  
3 -5 -6 -2 4  
-2 0 1 2 3  
-2 -4 1 10 21  
-1 0 1 2 3  
9 5 4 7 15  
-2 -1 0 0,5 1  
18 6 1 2 4  
-3 -2 -1 0 1  
0 -4 -6 -3 3  
-1 0 1 2 3  
8 1 -4 -7 -10  
-2 0 1 2 4  
18 3 1 3 21  
-3 -2 0 1 2  
-27 -14 -1 0 -3  
-3 -1 0 2 3  
10 -1 -3 -4 -1  
-2 -1 0 1 2  
22 4 -5 -9 -5  
-2 -1 0 1 2  
13 1 -2 3 14  
-2 0 1 2 3  
-7 -4 2 13 26  
-4 -1 0 2 3  
-11 -7 -2 14 27  
-2 -1 0 1 3  
-8 -10 -7 -2 24  
-2 -1 0 2 3  
15 3 -3 -3 3  
-1 0 1 2 3  
11 8 7 12 21  
0 1 2 3 4  
-9 -6 4 20 40  
-2 -1 0 1 3  
-12 -3 3 7 10  
-1 0 1 3 4  
12 6 6 26 42  
-3 -2 -1 0 3  
-5 -7 -5 -2 20  
-2 -1 1 2 3  
8 3 2 8 16  
-2 -1 0 1 2  
1 -5 -6 -5 0  
-3 -2 0 1 2  
-5 -1 3 1 -1  
-1 0 1 2 4  
0 1 4 12 37  
-2 -1 0 2 3  
8 6 4 9 12  
-2 -1 0 0,5 1  
-2 13 14 7 1  
-3 -2 -1 0 1  
-3 2 4 3 -3  
-1 0 1 2 3  
15 12 6 1 -2  
-2 0 1 2 4  
-3 6 7 3 -4  
-3 -2 0 1 2  
-2 6 20 14 -7  
-3 -1 0 2 3  
2 8 9 3 1  
-2 -1 0 1 2  
-10 -7 -4 1 8  
-2 -1 0 1 2  
-11 -5 3 3 3  
-2 0 1 2 3  
-5 -1 -1 1 3  
-4 -1 0 2 3  
6 12 13 8 6  
-2 -1 0 1 3  
1 5 6 4 2  
-2 -1 0 2 3  
-10 -8 -7 -2 24  
-1 0 1 2 3  
4 -9 -6 20 40  
0 1 2 3 4  
4 8 12 11 9  
-2 -1 0 1 3  
-6 0 3 2 -3  
-1 0 1 3 4  
-10 -3 0 -5 -8  
-3 -2 -1 0 3  
-10 -6 -5 -4 0  
-2 -1 1 2 3  
-4 -2 1 6 10  
-2 -1 0 1 2  
-28 -23 -20 -17 -12  
-3 -2 0 1 2  
-12 6 4 1 2  
-1 0 1 2 4  
-70 -28 -6 4 7  
-2 -1 0 2 3  
-7 6 8 10 5  
-2 0 1 2 3  
31 3 24 -34 -96  
-3 -1 0 1 2  
-18 3 -2 -3 2  
-2 0 1 2 3  
-6 1 0 -5 -5  
-2 0 2 3 4  
1 18 21 3 3  
-3 -2 0 1 2  
2 1 1 -9 -34  
-4 -2 0 1 2  
-6 3 4 5 -2  
-2 0 1 2 3  
4 9 12 7 5  
-1 0 1 2 3  
9 1 -1 -6 1  
-3 -2 0 1 2  
1 -4 -7 3 8  
-1 0 1 2 4  
-11 0 4 14 41  
-2 -1 0 2 3  
-14 -1 0 -3 -27

Файл f\_function.txt:

4052 4999 5403 5625 5764 5889 5928 5981 6022 6056 6082 6106  
98,49 99,01 99,17 99,25 99,3 99,33 99,34 99,36 99,38 99,4 99,41 99,42  
34,12 30,81 29,46 28,71 28,24 27,91 27,67 27,49 27,34 27,23 27,13 27,05  
21,2 18 16,69 15,98 15,52 15,21 14,98 14,8 14,66 14,54 14,45 14,37  
16,26 13,27 12,06 11,39 10,97 10,67 10,45 10,27 10,15 10,05 9,96 9,89  
13,74 10,92 9,78 9,15 8,75 8,47 8,26 8,1 7,98 7,87 7,79 7,72  
12,25 9,55 8,45 7,85 7,46 7,19 7 6,84 6,71 6,62 6,54 6,47  
11,26 8,65 7,59 7,01 6,63 6,37 6,19 6,03 5,91 5,82 5,74 5,67  
10,56 8,02 6,99 6,42 6,06 5,8 5,62 5,47 5,35 5,26 5,18 5,11  
10,04 7,56 6,55 5,99 5,64 5,39 5,21 5,06 4,95 4,85 4,78 4,71  
9,86 7,2 6,22 5,67 5,32 5,07 4,88 4,74 4,63 4,54 4,46 4,4  
9,33 6,93 5,95 5,41 5,06 4,82 4,65 4,5 4,39 4,3 4,22 4,16  
9,07 6,7 5,74 5,2 4,86 4,62 4,44 4,3 4,19 4,1 4,02 3,96  
8,86 6,51 5,56 5,03 4,69 4,46 4,28 4,14 4,03 3,94 3,86 3,8  
8,68 6,36 5,42 4,89 4,56 4,32 4,14 4 3,89 3,8 3,73 3,67  
8,53 6,23 5,29 4,77 4,44 4,2 4,03 3,89 3,78 3,69 3,61 3,55  
8,4 6,11 5,18 4,67 4,34 4,1 3,93 3,79 3,68 3,59 3,52 3,45

Файл t\_function.txt:

63,7  
9,92  
5,84  
4,6  
4,03  
3,71  
3,5  
3,36  
3,25  
3,17  
3,11  
3,05  
3,01  
2,98  
2,95  
2,92  
2,9  
2,88  
2,86  
2,85  
2,83  
2,82  
2,81  
2,8  
2,79  
2,78  
2,77  
2,76  
2,76  
2,75  
2,7

Файл task.txt:

12