|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК4 «Программное обеспечение ЭВМ,** |
| **информационные технологии»** | |

**Домашняя работа №1**

**«Первичная обработка данных»**

**ДИСЦИПЛИНА: «Методы обработки информации»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-72Б | |  |  | ( | Сафронов Н.С. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Никитенко У.В. | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

Калуга, 2023

**Цель:** формирование у студентов практических навыков обработки статистических данных.

**Задачи:** моделирование непрерывной СВ, анализ исходных данных, построение оценок плотности вероятности, нахождение точечных и интервальных оценок параметров распределения.

**Задание 1**

**Постановка задачи**

**Вариант №6**

|  |  |  |
| --- | --- | --- |
| **6** | **Накагами** |  |

1. Выполнить статистическое моделирование случайной величины с заданным законом распределения (табл. 1) путем генерации отсчетов α1i, i = 1, …, N случайных величин с 6 равномерным распределением в интервале [0, 1] (или, при необходимости нескольких СВ (α1, α2, ..., αk); N=10000. Сформировать соответствующий script-файл в среде MATLAB.

2. Получить гистограмму для закона распределения в соответствии с вариантом задания. Гистограмма может быть получена в среде MATLAB с помощью оператора hist(X1,N), X1 — анализируемая случайная величина, N — число интервалов на гистограмме, которое должно составлять от 100 до 500. Сравнить полученную гистограмму с соответствующим графиком плотности вероятности f(x) в соответствии с заданием.

3. Вычислить: − выборочное среднее значение, − медиану, − нижний и верхний квартиль, − выборочную дисперсию и СКО, смоделированной случайной величины и сравнить их с теоретическими значениями (мат. ожиданием и дисперсией, медианой, нижним и верхним квартилем).

4. Сделать выводы.

**Листинг программы:**

% Вариант 6

% Накагами

% sigma=1, m=0.5

sigma = 1;

m = 0.5;

func = @(x) ((2\*m^m)/(gamma(m)\*sigma^m))\*x^(2\*m-1)\*exp(-((m/sigma)\*x^2)) \* 200;

size = 5000;

N = 100;

xs = randn(1, size);

x\_range = linspace(0, 1, N);

y\_range = linspace(0, 1, N);

for i=1:N

y\_range(1, i) = func(x\_range(1, i));

end

alpha = nakagami\_dist(m, sigma, size);

histogram(alpha, N);

hold on;

plot(x\_range, y\_range);

meanValue = mean(alpha); % Выборочное среднее

medmnValue = median(alpha); % Медиана

lowerQuantile = quantile(alpha, 0.25); % Верхний квартиль

upperQuantile = quantile(alpha, 0.75); % Верхний квартиль

varmnce = var(alpha); % Выборочная дисперсия

stdDevmtion = std(alpha); % Стандартное отклонение

dist = makedist('Nakagami', m, sigma);

fprintf('-------МОДЕЛЬ-------\n');

fprintf('Выборочное среднее значение: %.4f\n', meanValue);

fprintf('Медиана: %.4f\n', medmnValue);

fprintf('Нижний квартиль: %.4f\n', lowerQuantile);

fprintf('Верхний квартиль: %.4f\n', upperQuantile);

fprintf('Выборочная дисперсия: %.4f\n', varmnce);

fprintf('Стандартное отклонение: %.4f\n', stdDevmtion);

fprintf('-------ТЕОРИЯ-------\n');

fprintf('Выборочное среднее значение: %.4f\n', mean(dist));

fprintf('Медиана: %.4f\n', median(dist));

fprintf('Нижний квартиль: %.4f\n', icdf(dist, 0.25));

fprintf('Верхний квартиль: %.4f\n', icdf(dist, 0.75));

fprintf('Дисперсия: %.4f\n', var(dist));

fprintf('Стандартное отклонение: %.4f\n', std(dist));

function x = gamma\_dist(alpha, beta, n)

x = zeros(n,1);

for i = 1:n

while true

u1 = rand();

u2 = rand();

v1 = log(u1)/alpha;

v2 = log(u2);

if alpha < 1

w = u2^(1/alpha);

else

w = exp(-v2\*(alpha-1));

end

if w <= (1 - alpha\*v1)

x(i) = w^(1/alpha)\*beta;

break;

end

end

end

end

function x = nakagami\_dist(m, sigma, n)

x = zeros(n,1);

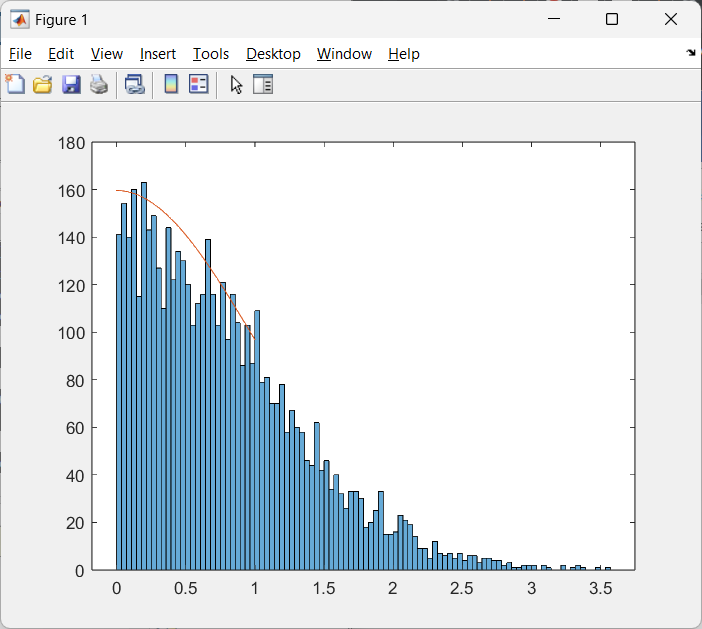
for i = 1:n

x(i) = sqrt(gamrnd(m, sigma/m, 1)/(2\*m)\*sigma);

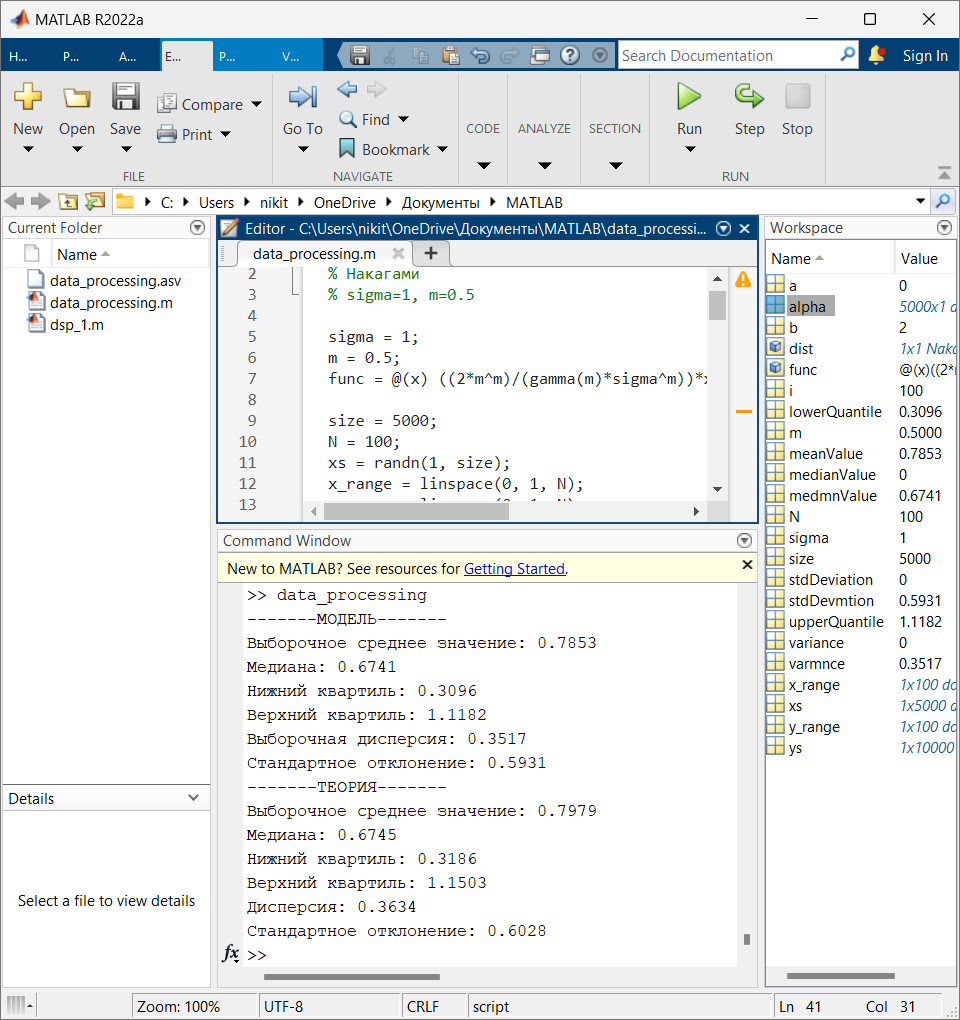
end

end

**Результат:**



**Рисунок 1** –Графики теоретической и эксперементальной



**Рисунок 2** –Полученные характеристики теоретического и эксперементального распределения

**Задание 2**

**Вариант 14**

**Постановка задачи**

Для обработки преподавателем выдается случайных чисел.

Эти числа хранятся в файле TestNN.csv.

1. Выборка подвергается обработке и оформляется в виде таблицы.

2. Графические характеристики выборки – строим гистограмму и полигон приведенных частот. Выдвигаем гипотезу о виде плотности вероятности генерального распределения.

3. Находим выборочные характеристики положения и рассеивания.

4. Для сравнения с гистограммой и полигоном приведенных частот на одном чертеже постройте графики гистограммной оценки плотности вероятности параметрической oцeнки плотности вероятности , и усредненную ядерную оценку плотности вероятности .

5. Значения оценок плотности вероятности в средних точках промежутков группированного статистического ряда оформите в виде таблицы.

6. Проанализируйте близость оценок по средним квадратическим отклонениям и от .

**Листинг программы**

import argparse

import csv

import numpy as np

import prettytable

import matplotlib.pyplot as plt

import statistics as st

from scipy.stats import gaussian\_kde

if \_\_name\_\_ == "\_\_main\_\_":

parser = argparse.ArgumentParser()

parser.add\_argument("-file") or "./data/Test14.csv"

args = parser.parse\_args()

file = args.file

points = []

with open(file, newline='') as csvfile:

reader = csv.reader(csvfile, delimiter=' ', quotechar='|')

for row in reader:

points.append(float("".join(row)))

points.sort()

min\_point = points[0]

max\_point = points[-1]

points\_range = max\_point - min\_point

print(f"Размах выборки: {points\_range:.2f}")

num\_bins = 1 + int(np.ceil(np.log2(len(points))))

print(f"Количество интервалов: {num\_bins}")

step = points\_range / num\_bins

print(f"Длина интервала: {step:.2f}")

\_bins = []

for i in range(num\_bins):

current\_min = min\_point + i \* step

current\_max = min\_point + (i + 1) \* step

current\_range = (current\_min, current\_max)

count = len(

list(

filter(

lambda x: current\_min <= x <= current\_max, points

)

)

)

current\_average = (current\_max + current\_min) / 2.

\_bins.append(

{

"average": round(current\_average, 4),

"minimum": round(current\_min, 4),

"maximum": round(current\_max, 4),

"count": round(count, 4)

}

)

table = prettytable.PrettyTable()

table.field\_names = [

"Номер промежутка", "a\_{i-1}", "a\_i", "n\_i",

"Средняя точка промежутка"

]

index = 1

for bin in \_bins:

table.add\_row([index := index + 1, bin["minimum"], bin["maximum"],

bin["count"], bin["average"]])

print(table)

print(f"Выборочное среднее: {np.mean(points):.2f}")

print(f"Медиана: {np.median(points):.2f}")

print(f"Мода: {st.mode(points):.2f}")

print(f"Размах выборки: {max(points) - min(points):.2f}")

print(f"Выборочная дисперсия: {np.var(points):.2f}")

print(f"Стандартное отклонение выборки: {np.sqrt(np.var(points)):.2f}")

print(f"Коэффициент вариации: "

f"{np.sqrt(np.var(points)) / np.mean(points) / 100:.2f}")

plt.hist(

points, color="grey", edgecolor="black", bins=num\_bins, range=(

min\_point,

max\_point), alpha=0.5, density=True, label="Гистограмма"

)

centers = [bin["average"] for bin in \_bins]

\_bins\_plot = [bin["count"] / 49 for bin in \_bins]

plt.plot(centers, \_bins\_plot, color="black")

plt.show()

def normal\_distribution(x):

return 1 / np.sqrt(2 \* np.pi) / np.sqrt(np.var(points)) \* (

np.exp(-1 / 2 \* (

(x - np.mean(points)) / np.sqrt(np.var(points))) \*\* 2

)

)

kde = gaussian\_kde(points)

xs = np.linspace(min\_point, max\_point, 100)

plt.hist(

points, color="grey", edgecolor="black", bins=num\_bins, range=(

min\_point,

max\_point), alpha=0.5, density=True, label="Гистограмма"

)

plt.plot(xs, [kde(x) for x in xs], color="red",

label="Усреднённая ядерная оценка")

plt.plot(xs, [normal\_distribution(x) for x in xs], color="green",

label="Параметрическая оценка")

plt.legend()

plt.show()

table = prettytable.PrettyTable()

table.field\_names = [

"z\_i", "n\_i", "f\_Г(x)", "f\_УЯ(x)", "f\_П(x)", "(f\_УЯ(x)-f\_Г(x))^2",

"(f\_П(x)-f\_Г(x))^2",

]

index = 1

for bin in \_bins:

current\_average = bin["average"]

current\_count = bin["count"]

current\_histogram = round(current\_count / len(points) / 0.5, 4)

current\_kde = round(float(kde(current\_average)), 4)

current\_parametric = round(normal\_distribution(current\_average), 4)

diff\_kde = round((current\_kde - current\_histogram) \*\* 2, 4)

diff\_parametric = round((current\_parametric - current\_histogram) \*\* 2, 4)

table.add\_row([

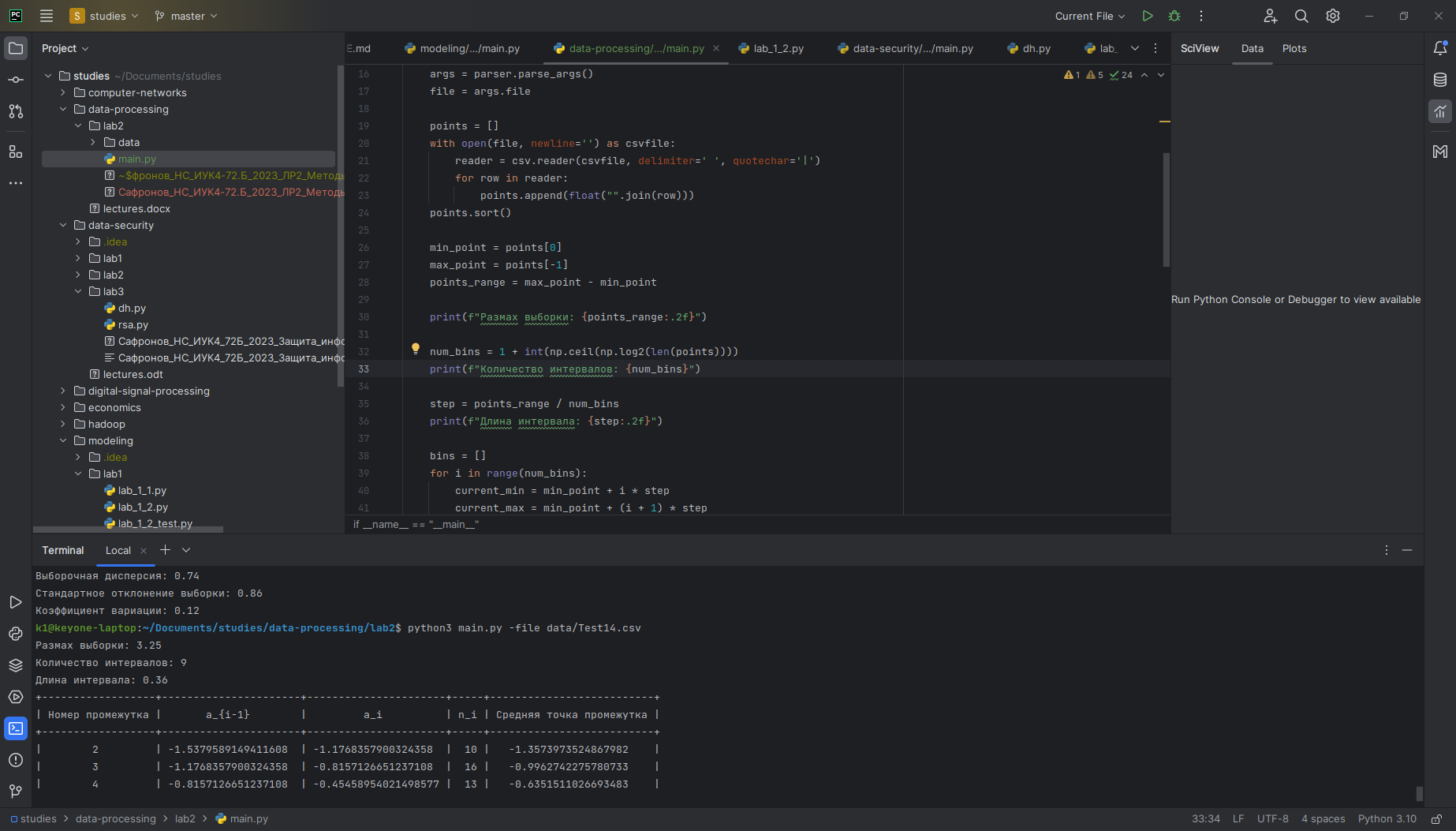
current\_average, current\_count, current\_histogram,

current\_kde, current\_parametric, diff\_kde, diff\_parametric

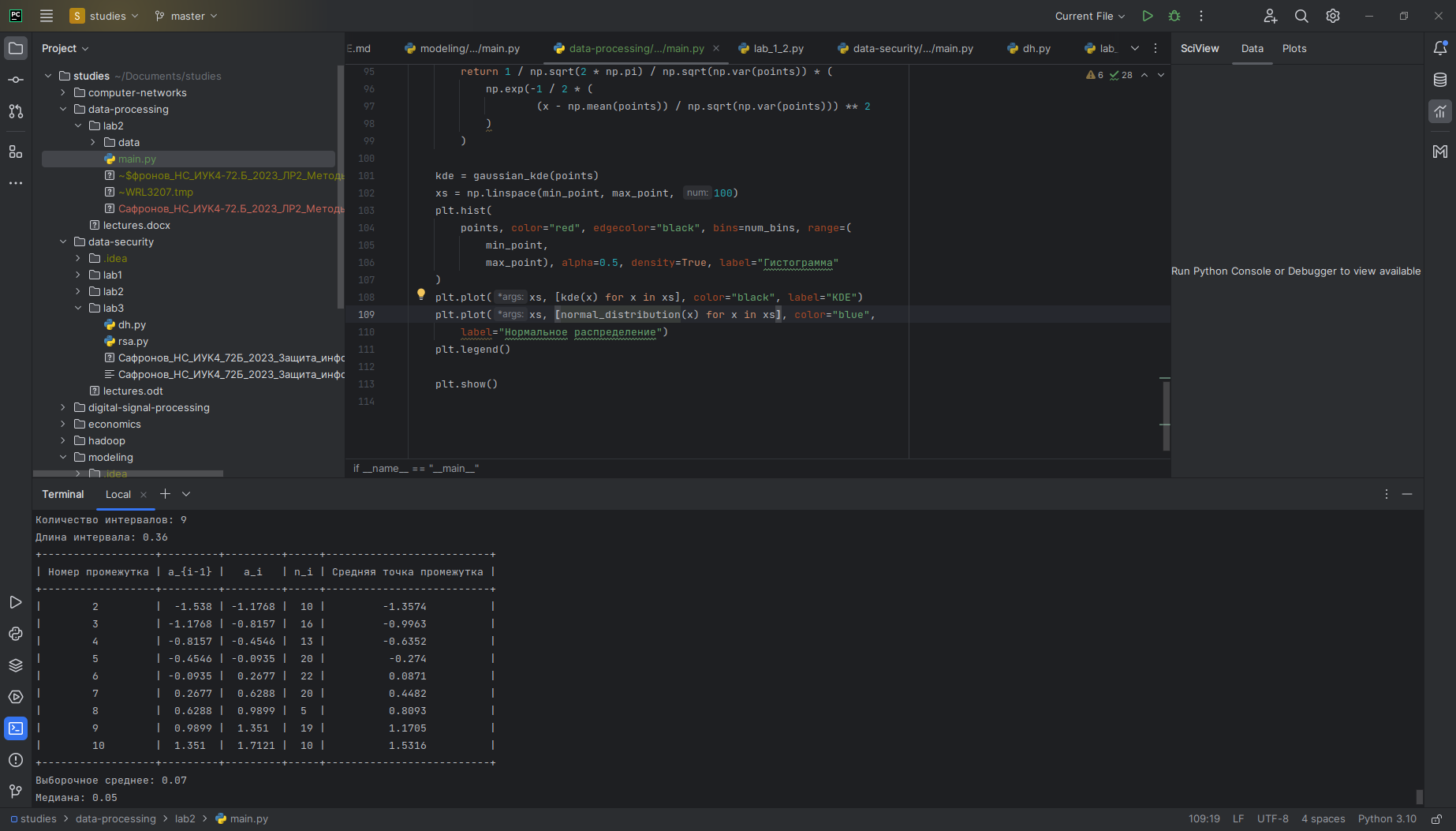
])

print(table)

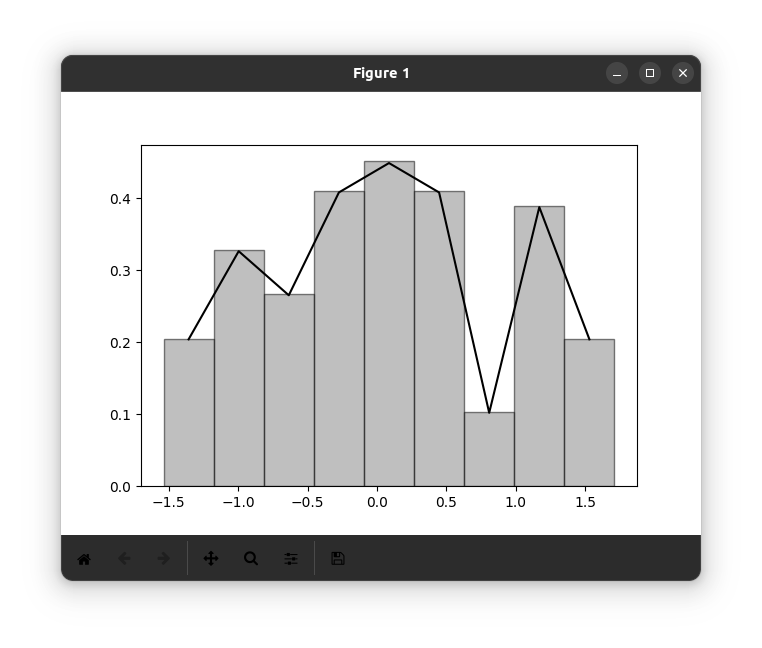
**Результаты выполнения программы**

****

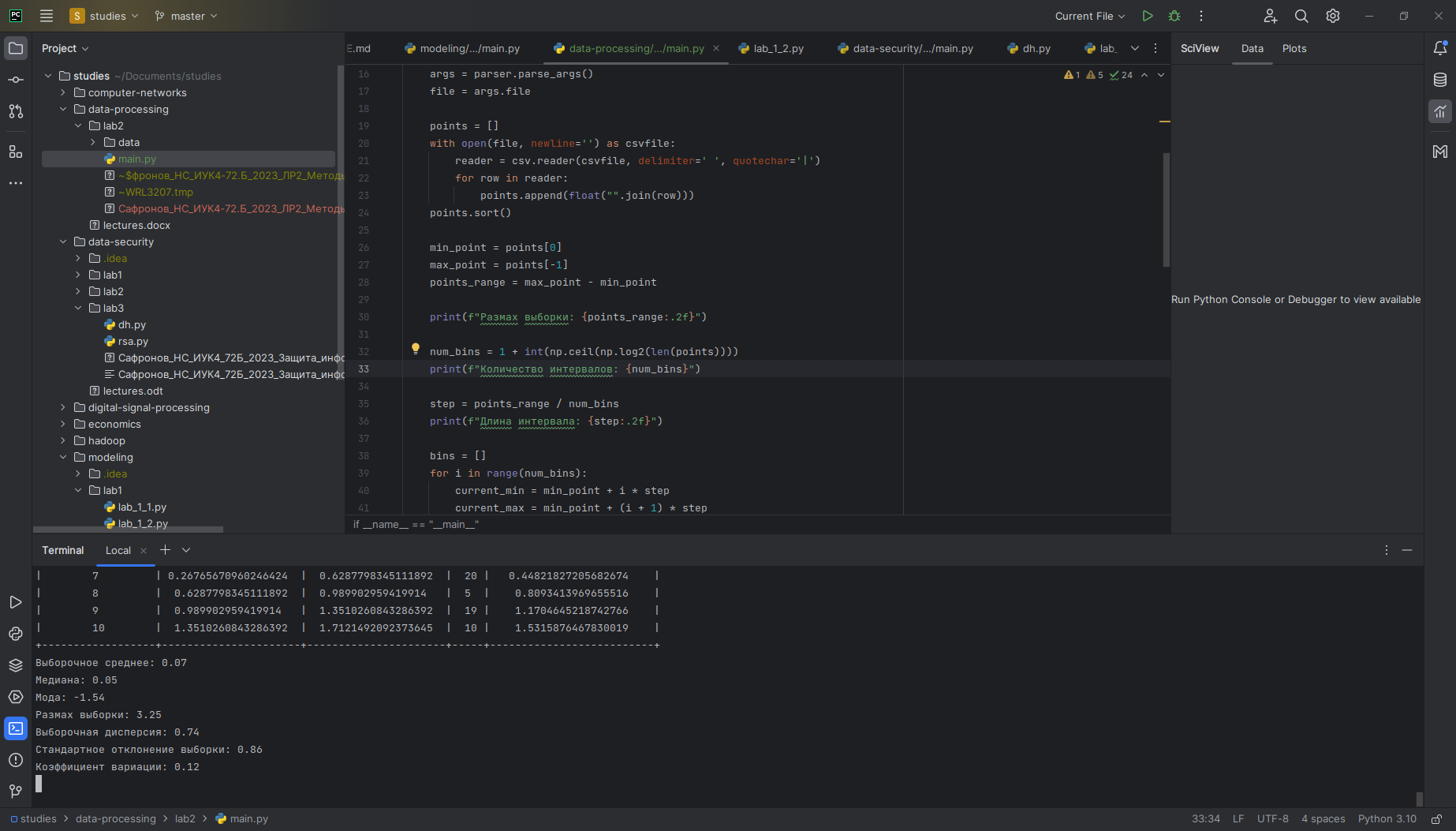
**Рисунок 3 –** Параметры построения гистограммы



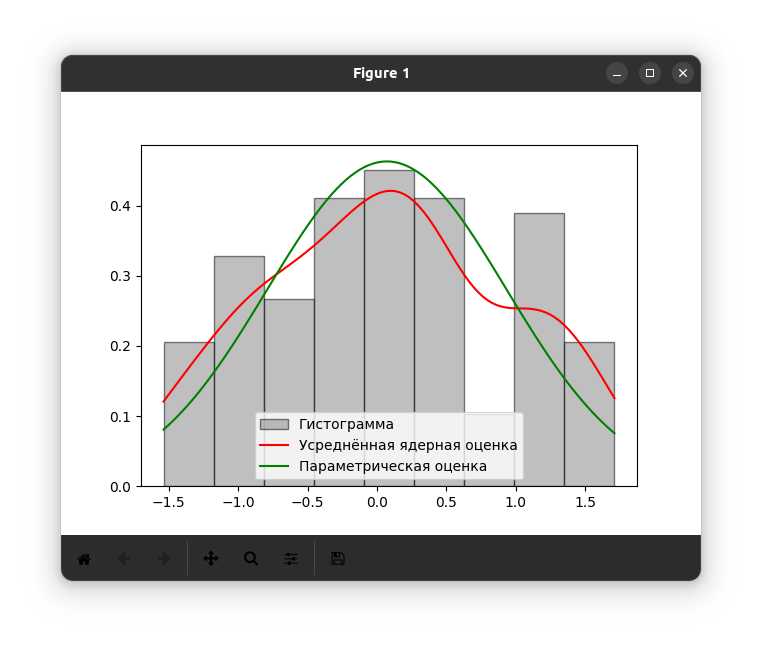
**Рисунок 4 –** Результат обработки выборки

****

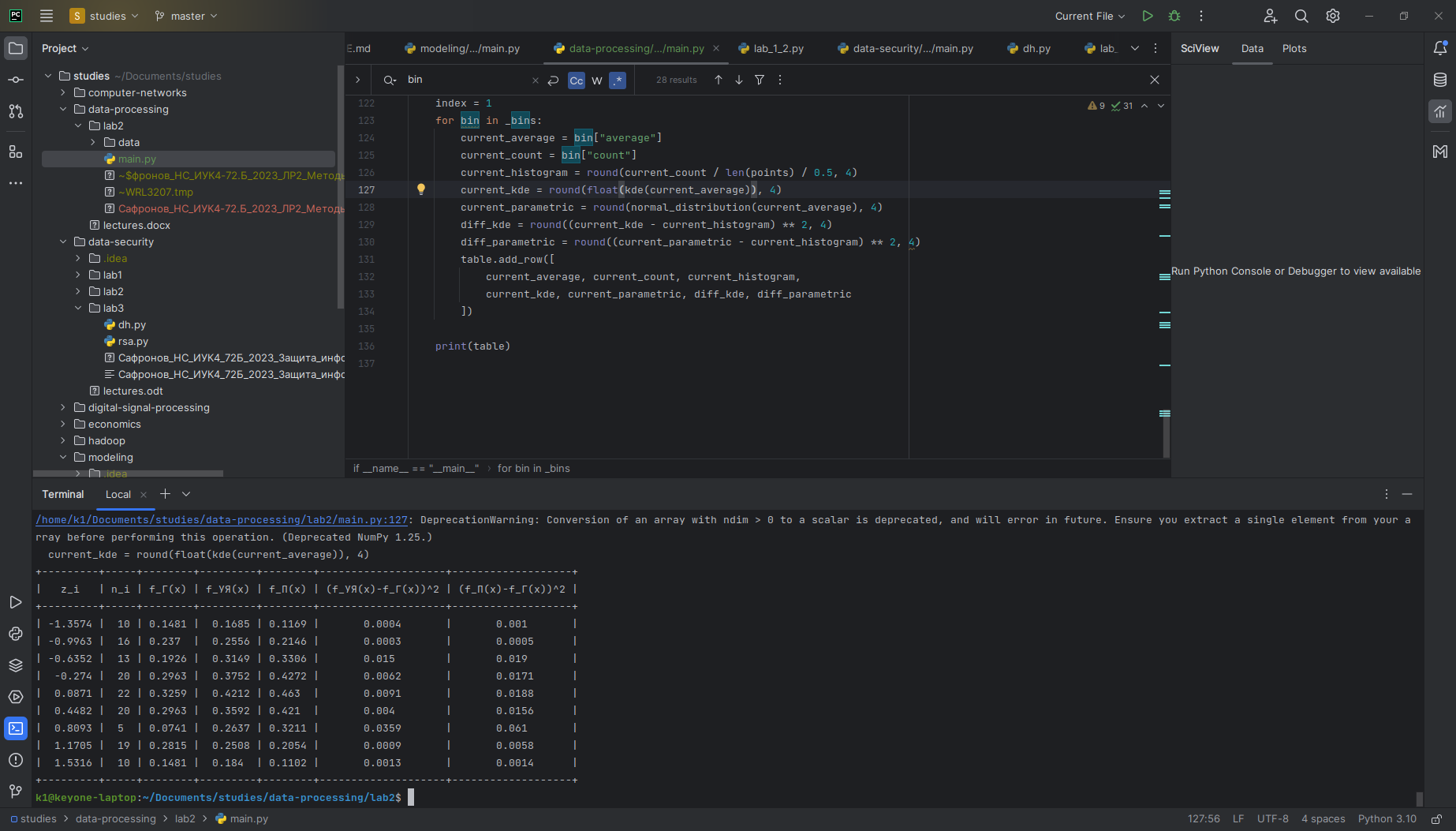
**Рисунок 5 –** Построенные гистограмма и полигон частот



**Рисунок 6 –** Выборочные характеристики положения и рассеяния



**Рисунок 7 –** Параметрическая и усреднённая ядерная оценки выборки



**Рисунок 8 –** Значения плотностей вероятности в средних точках интервалов

Усреднённая ядерная оценка находиться ближе к гистограммной оценке плотности, поскольку в отличие от параметрической она является ассиметричной.

**Задание 3**

**Постановка задачи**

Сгенерировать выборку из 100 элементов, имеющих указанное в вашем варианте распределение. Считая один из параметров распределения неизвестным, найти его точечную оценку:

а) методом моментов (c помощью указанных в задании моментов);

б) методом максимального правдоподобия.

Построить график функции правдоподобия и убедиться, что найденная с помощью метода максимального правдоподобия оценка действительно является точкой максимума функции правдоподобия. Сравнить полученные точечные оценки с истинным значением параметра распределения.

**Вариант 14**

X - выборка из распределения , где k = 3. Найти оценку параметра k, считая его неизвестным. Метод моментов реализовать с помощью моментов 1-го и 2-го порядков.

**Ход выполнения практического задания**

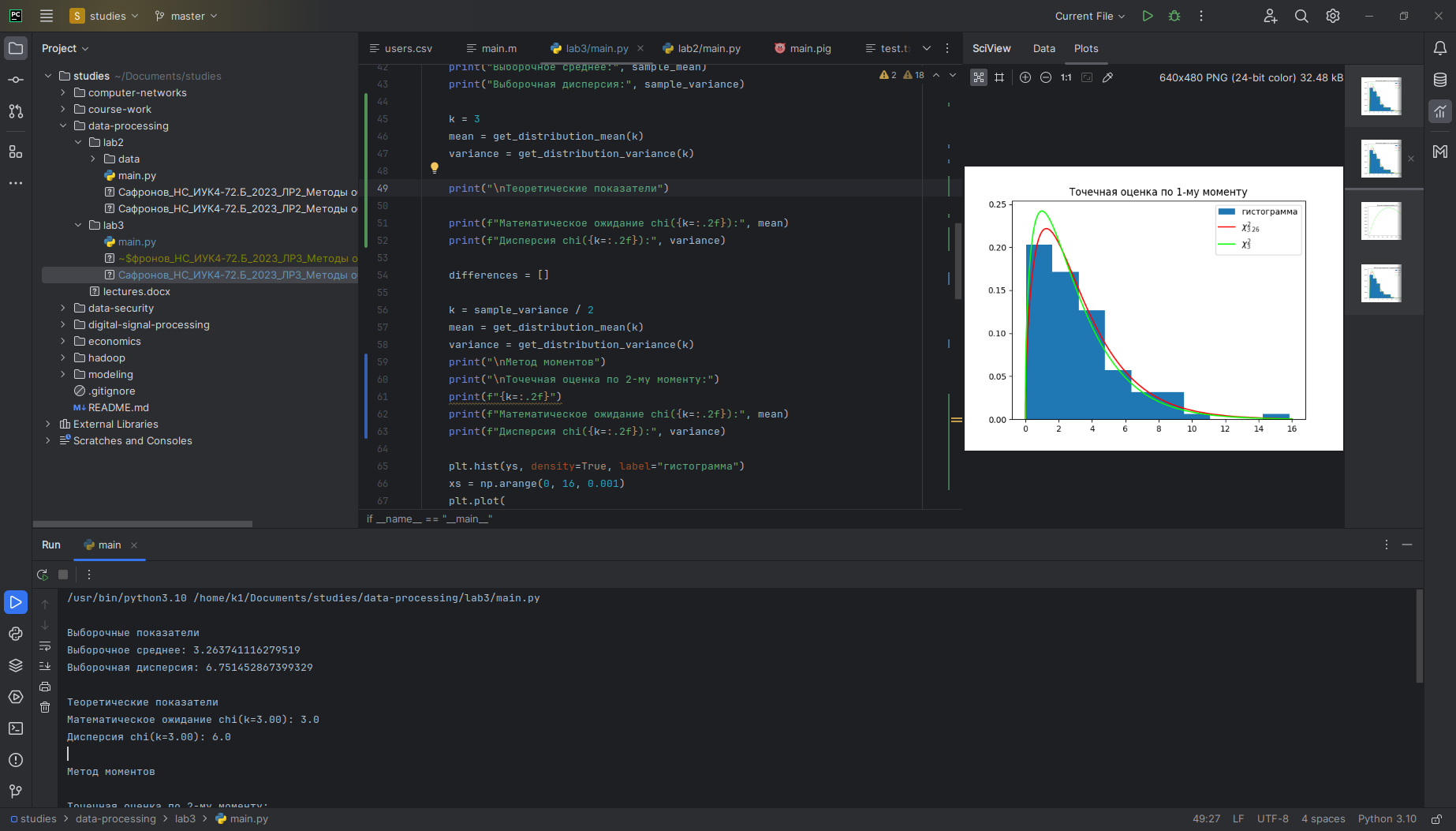
Выпишем формулы для нахождения математического ожидания и дисперсии для распределения :

Получаем следующие точечные оценки для :

Для момента 1-го порядка:

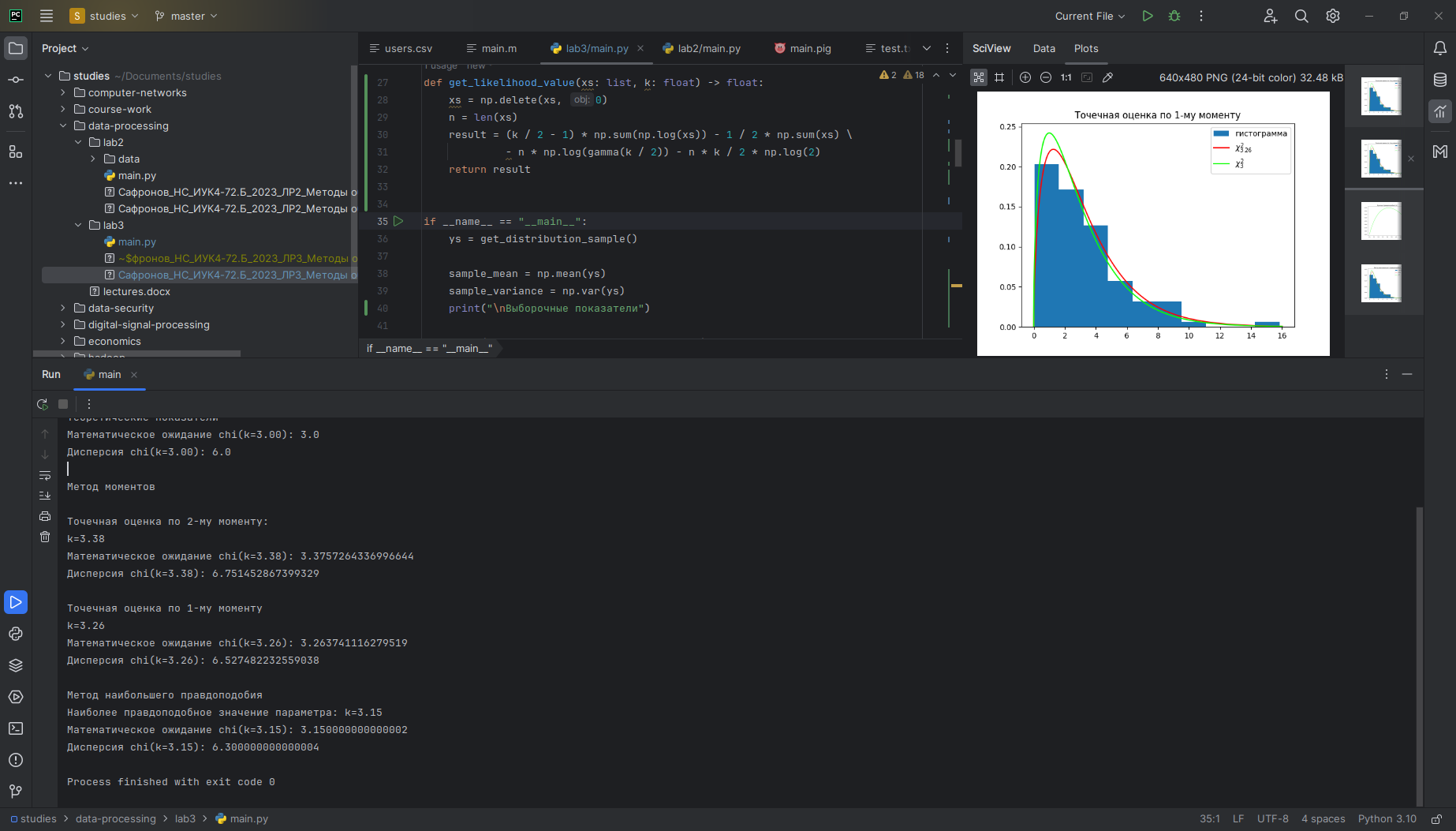
Для момента 2-го порядка:

Найдём выборочные характеристики распределения:



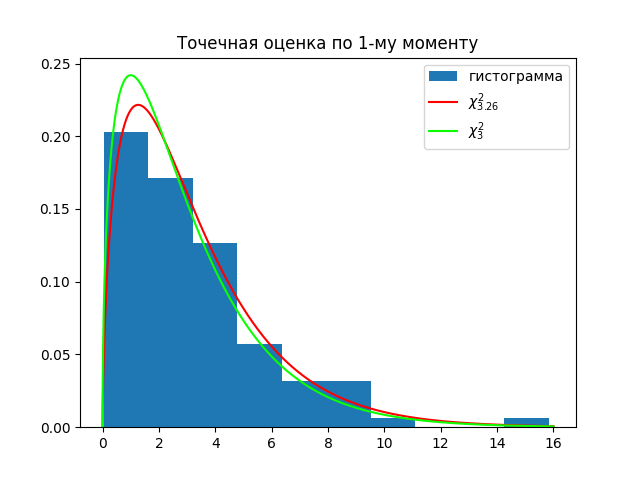
**Рисунок 8 –** Выборочные и теоретические показатели распределения

Воспользовавшись методом моментов, найдём точечную оценку параметра :

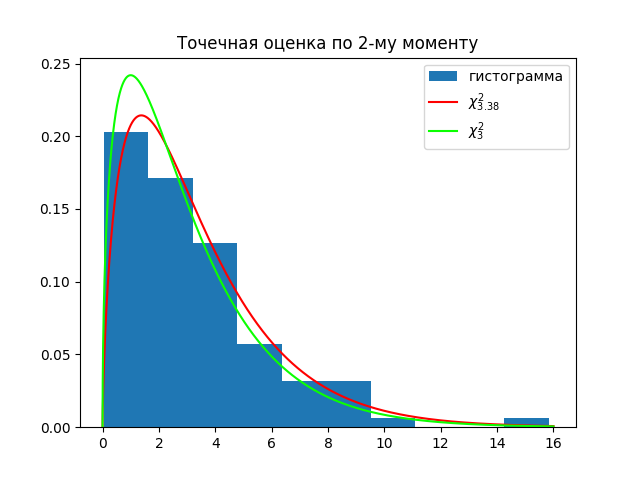


**Рисунок 10 –** Точечные оценки параметра, полученные методом моментов

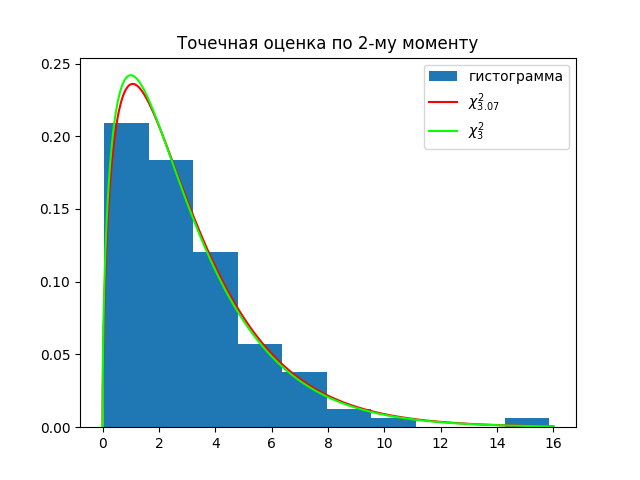
Построим графики, соответствующие полученным значениям параметра:



**Рисунок 11 –** График функции при точечной оценке, полученной по первому моменту

****

**Рисунок 12** – График функции при точечной оценке, полученной по второму моменту

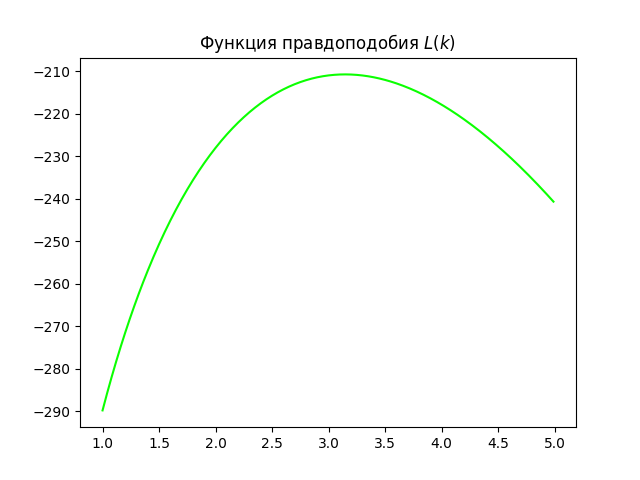
****

**Рисунок 13** – Точечная оценка параметра , вычисленная методом моментов 2-го порядка

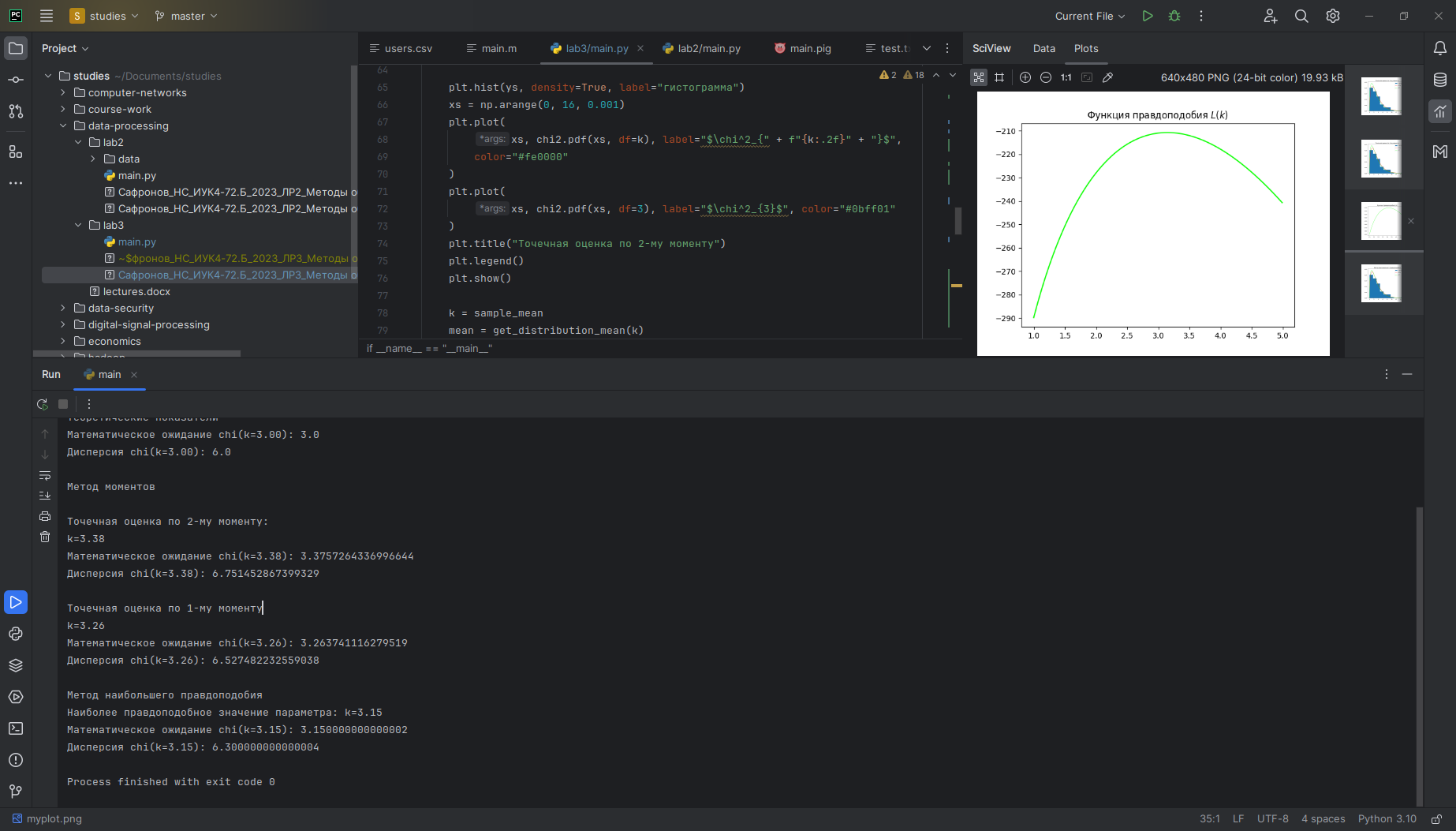
Воспользуемся методом максимального правдоподобия.

Построим логарифмическую функцию правдоподобия для заданного распределения:

Построим график зависимости логарифмической функции правдоподобия на заданном промежутке значений при заданных значениях выборки. Найдём максимальное значение функции и точку, соответствующую ему.

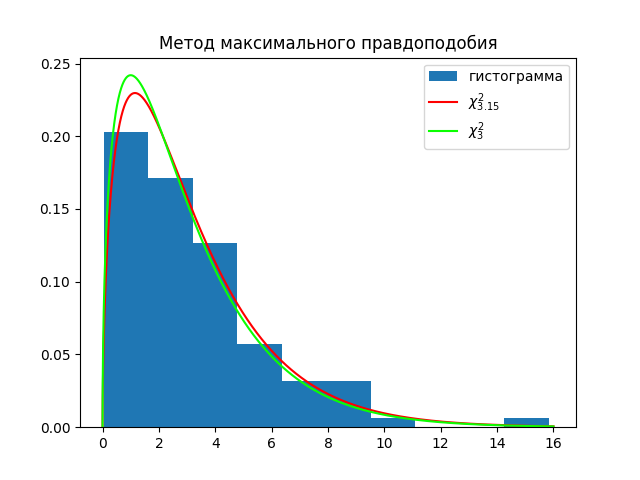
****

**Рисунок 14 –** Функция правдоподобия



**Рисунок 15 –** Точечная оценка параметра, полученная методом максимального правдоподобия

Построим график, соответствующий полученному значению параметра:



**Рисунок 16 –** График функции при точечной оценке, полученной методом максимального правдоподобия

Таким образом, получаем, что наиболее точной оказалась оценка, полученная методом моментов по второму моменту.

**Задание 4**

**Постановка задачи**

1. Для обеих выборок построить точный доверительный интервал уровня доверия для параметра , считая:

а) a неизвестным,

б) a известным и равным .

1. В одной системе координат построить графики зависимости длины доверительного интервала от уровня доверия для всех четырех случаев (объем выборки равен , a неизвестно; объем выборки равен , a известно; объем выборки равен , a неизвестно; объем выборки равен , a известно). При этом q придать минимум 50 разных значений через равные промежутки.

**Вариант 14**

**Ход выполнения практического задания**

Примем размеры малой и большой выборок , соответственно.

Формула доверительного интервала для σ2 при известном а:

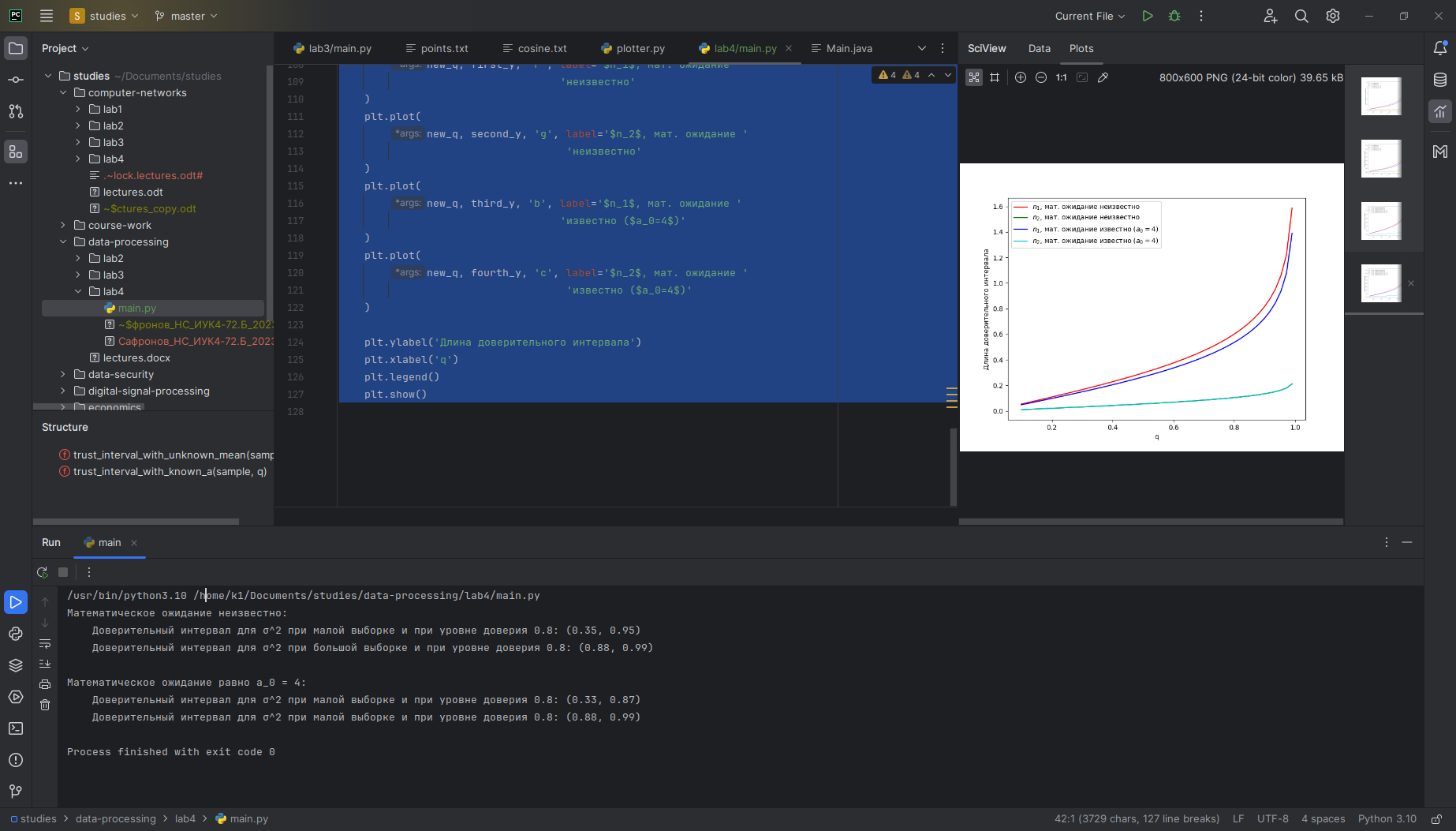
*, где*

*.*

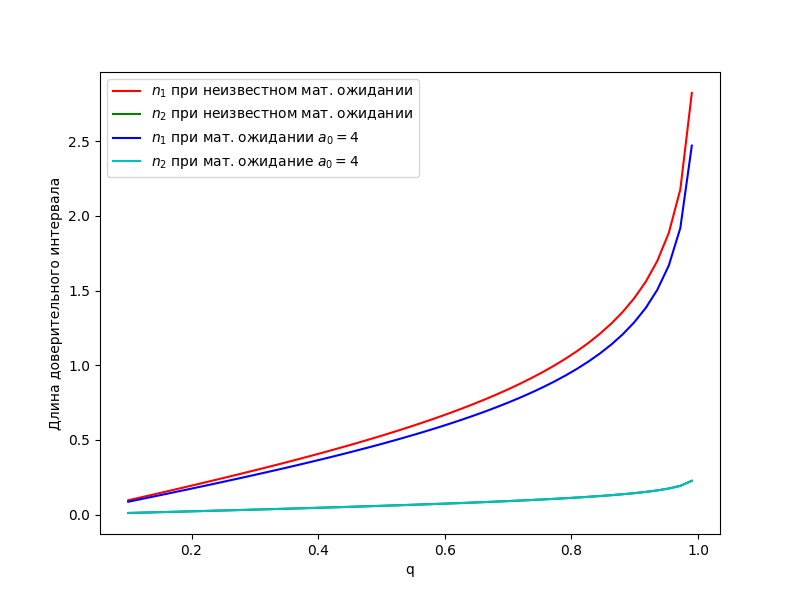
Формула доверительного интервала для σ2 при неизвестном а:

*, где*

*.*



**Рисунок 17 –** Полученные доверительные интервалы



**Рисунок 18 -** Графики зависимости длины доверительного интервала от уровня доверия

Длина доверительного интервала, характеризующая точ­ность интервального оценивания, зависит от объема выборки *n* и уровня доверия: при увеличении объема выборки длина доверительного интервала уменьшается, а при приближении уровня доверия к единице − увеличивается. Также при известном значении *a* и в случае большой выборки, длина доверительного интервала стабильна независимо от уровня доверия.

**Листинг программы**

import matplotlib.pyplot as plt  
import numpy as np  
import scipy.stats as stats  
  
  
def trust\_interval\_with\_unknown\_mean(  
 sample: np.array,  
 q: float  
) -> tuple[float, float]:  
  
 alpha = 1 - q  
 data = np.array(sample)  
 n = len(sample)  
 sample\_variance = np.var(data, ddof=1)  
  
 chi2\_lower = stats.chi2.ppf(alpha / 2, df=n - 1)  
 chi2\_upper = stats.chi2.ppf(1 - alpha / 2, df=n - 1)  
  
 lower\_bound = (n - 1) \* sample\_variance / chi2\_upper  
 upper\_bound = (n - 1) \* sample\_variance / chi2\_lower  
  
 return lower\_bound, upper\_bound  
  
  
def trust\_interval\_with\_known\_a(  
 sample: np.array,  
 q: float  
) -> tuple[float, float]:  
  
 alpha = 1 - q  
 sample\_variance = np.var(sample, ddof=0)  
 degrees\_of\_freedom = n = len(sample)  
  
 chi2\_lower = stats.chi2.ppf(alpha / 2, df=degrees\_of\_freedom)  
 chi2\_upper = stats.chi2.ppf(1 - alpha / 2, df=degrees\_of\_freedom)  
  
 lower\_bound = (n \* sample\_variance) / chi2\_upper  
 upper\_bound = (n \* sample\_variance) / chi2\_lower  
  
 return lower\_bound, upper\_bound  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 mean, sigma = 4, 1  
 q = 0.8  
  
 first\_count = 15  
 second\_count = first\_count \* 70  
  
 first\_sample = np.random.normal(mean, sigma, first\_count)  
 second\_sample = np.random.normal(mean, sigma, second\_count)  
  
 print('Математическое ожидание неизвестно:')  
  
 lower\_bound\_SPA, upper\_bound\_SPA = trust\_interval\_with\_unknown\_mean(  
 first\_sample, q  
 )  
 print(  
 "\tДоверительный интервал для σ^2 при малой выборке и при уровне "  
 f"доверия {q}: ({lower\_bound\_SPA:.2f}, {upper\_bound\_SPA:.2f})"  
 )  
  
 lower\_bound\_BPA, upper\_bound\_BPA = trust\_interval\_with\_unknown\_mean(  
 second\_sample, q  
 )  
 print(  
 "\tДоверительный интервал для σ^2 при большой выборке и при уровне "  
 f"доверия {q}: ({lower\_bound\_BPA:.2f}, {upper\_bound\_BPA:.2f})"  
 )  
  
 print(f"\nМатематическое ожидание равно a\_0 = {mean}:")  
 lower\_bound\_SPB, upper\_bound\_SPB = trust\_interval\_with\_known\_a(  
 first\_sample, q  
 )  
 print(  
 "\tДоверительный интервал для σ^2 при малой выборке и при уровне доверия "  
 f"{q}: ({lower\_bound\_SPB:.2f}, {upper\_bound\_SPB:.2f})"  
 )  
  
 lower\_bound\_BPB, upper\_bound\_BPB = trust\_interval\_with\_known\_a(  
 second\_sample, q  
 )  
 print(  
 "\tДоверительный интервал для σ^2 при малой выборке и при уровне доверия "  
 f"{q}: ({lower\_bound\_BPB:.2f}, {upper\_bound\_BPB:.2f})"  
 )  
  
 new\_q = np.linspace(0.1, 0.99, 50)  
 first\_y = []  
 second\_y = []  
 third\_y = []  
 fourth\_y = []  
  
 for i in range(50):  
 left, right = trust\_interval\_with\_unknown\_mean(first\_sample, new\_q[i])  
 first\_y.append(right - left)  
 left, right = trust\_interval\_with\_unknown\_mean(second\_sample, new\_q[i])  
 second\_y.append(right - left)  
 left, right = trust\_interval\_with\_known\_a(first\_sample, new\_q[i])  
 third\_y.append(right - left)  
 left, right = trust\_interval\_with\_known\_a(second\_sample, new\_q[i])  
 fourth\_y.append(right - left)  
  
 plt.figure(figsize=(8, 6))  
  
 plt.plot(  
 new\_q, first\_y, 'r', label='$n\_1$, мат. ожидание '  
 'неизвестно'  
 )  
 plt.plot(  
 new\_q, second\_y, 'g', label='$n\_2$, мат. ожидание '  
 'неизвестно'  
 )  
 plt.plot(  
 new\_q, third\_y, 'b', label='$n\_1$, мат. ожидание '  
 'известно ($a\_0=4$)'  
 )  
 plt.plot(  
 new\_q, fourth\_y, 'c', label='$n\_2$, мат. ожидание '  
 'известно ($a\_0=4$)'  
 )  
  
 plt.ylabel('Длина доверительного интервала')  
 plt.xlabel('q')  
 plt.legend()  
 plt.show()

**Вывод:** в ходе выполнения домашней работы были получены практические навыки обработки статистических данных.

**РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА**

1. Гмурман, В.Е. Теория вероятностей и математическая статистика: учеб. пособие для вузов/ В.Е. Гмурман. - М.: Юрайт, 2014. – 479 с. 21
2. Гринь, А.Г. Вероятность и статистика [Электронный ресурс]: учебное пособие/ А.Г. Гринь.— Омск: Омский государственный университет им. Ф.М. Достоевского, 2013.— 304 c.— Режим доступа: http://www.iprbookshop.ru/24879.html
3. Кельберт М.Я. Вероятность и статистика в примерах и задачах [Электронный ресурс]/ Кельберт М.Я. Сухов Ю.М.. - М.: МЦНМО, 2010. - Т. 1. Основные понятия теории вероятностей и математической статистики. - 486 с. - URL: //biblioclub.ru/index.php?page=book&id=69109