|  |  |
| --- | --- |
| **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 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Никитенко У.В. )  (Подпись) |
| 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:**

1. Выполнить статистическое моделирование случайной величины с заданным законом распределения путем генерации отсчетов α1i, i = 1, …, N случайных величин с 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).

1. Сделать выводы.

**Вариант 11**

* Закон распределения: Хи-квадрат
* Алгоритм: E2
* = 3
* = 2

**Листинг:**

import math, random, statistics

import matplotlib.pyplot as plt

import numpy as np

from scipy.stats import chi2

v = 3

sig = 2

G = math.pi / 2

f = lambda x: x\*\*((v - 2) / 2) / (2\*\*(v / 2) \* G \* sig\*\*v) \* math.exp(-x / (2 \* sig\*\*2))

N = 10000

s = [0] \* N

for i in range(N):

    for j in range(v):

        x = random.normalvariate(0, sig)

        s[i] += x\*\*2

print("--------------Модель--------------")

print(f"Выборочное среднее значение: {statistics.fmean(s):.3f}")

print(f"Медиана: {statistics.median(s):.3f}")

print(f"Верхний квартиль: {np.percentile(s, 75):.3f}")

print(f"Нижний квартиль: {np.percentile(s, 25):.3f}")

print(f"Выборочная дисперсия: {statistics.variance(s):.3f}")

print(f"СКО: {statistics.stdev(s):.3f}")

print("--------------Теория--------------")

print(f"Выборочное среднее значение: {chi2.mean(v):.3f}")

print(f"Медиана: {chi2.median(v):.3f}")

print(f"Верхний квартиль: {chi2.cdf(0.75, v):.3f}")

print(f"Нижний квартиль: {chi2.cdf(0.25, v):.3f}")

print(f"Выборочная дисперсия: {chi2.var(v):.3f}")

print(f"СКО: {chi2.std(v):.3f}")

X = np.linspace(0, 100, 101)

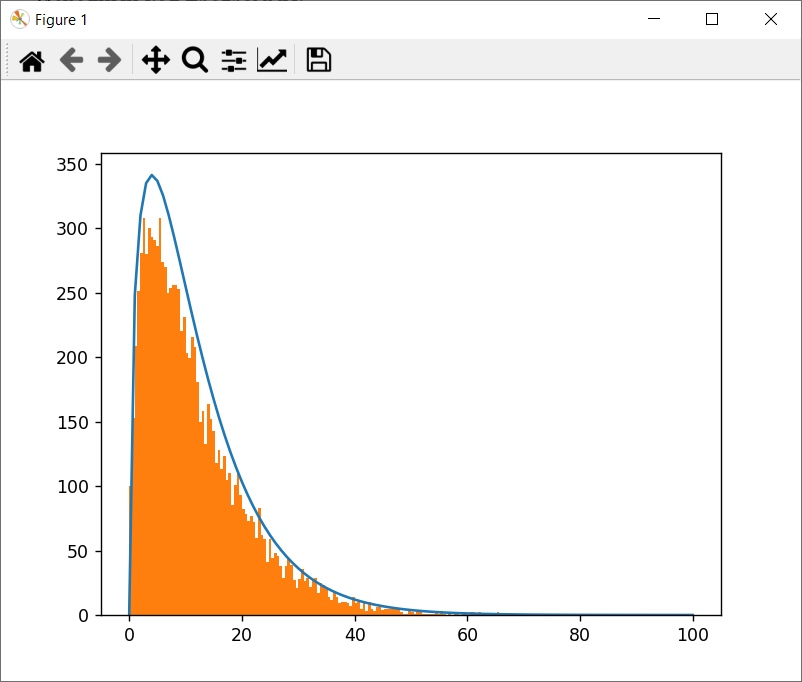
Y = [f(x) \* 10000 for x in X]

plt.plot(X, Y)

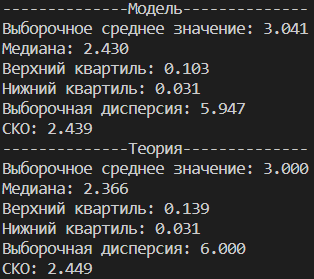
plt.hist(s, bins=200)

plt.show()

**Результат:**



**Рис. 1.** Графики

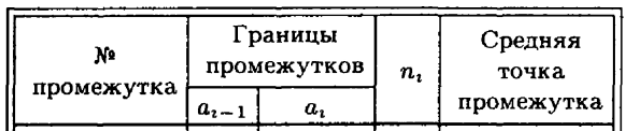


**Рис. 2.** Вычисления

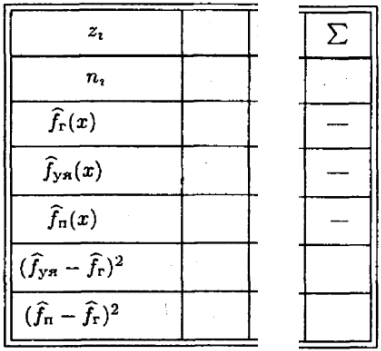
**Задание 2:**

Для обработки преподавателем выдается случайных чисел. Эти числа хранятся в файле TestNN.csv.

1. Выборка подвергается обработке и оформляется в виде таблицы.



1. Графические характеристики выборки – строим гистограмму и полигон приведенных частот. Выдвигаем гипотезу о виде плотности вероятности генерального распределения.
2. Находим выборочные характеристики положения и рассеивания.
3. 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 .
4. Значения оценок плотности вероятности в средних точках промежутков группированного статистического ряда оформите в виде таблицы.



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

**Вариант 7**

**Листинг:**

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import norm, gaussian\_kde

import pandas as pd

from prettytable import PrettyTable

data = pd.read\_csv('Test7.csv', header=None)

values = data.iloc[:, 0]

num\_bins = 8

plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)

plt.hist(values, bins=num\_bins, density=True, alpha=0.6, label='Гистограмма')

hist, bin\_edges = np.histogram(values, bins=num\_bins, density=True, range=(values.min(), values.max()))

bin\_centers = (bin\_edges[:-1] + bin\_edges[1:]) / 2

plt.plot(bin\_centers, hist, marker='o', linestyle='-', label='Полигон')

plt.title("Гистограмма и полигон частот")

plt.legend()

edges = [(round(bin\_edges[i], 2), round(bin\_edges[i + 1], 2)) for i in range(num\_bins)]

table = PrettyTable()

table.add\_column("#", [i for i in range(1, num\_bins + 1)])

table.add\_column("Промежуток", edges)

table.add\_column("Максимальное значение в этой точке", list(map(lambda value: round(value, 4), hist)))

table.add\_column("Центр промежутка", list(map(lambda center: round(center, 2), bin\_centers)))

print(table)

plt.subplot(1, 2, 2)

x = np.linspace(values.min(), values.max(), 100)

plt.plot(x, norm.pdf(x, loc=values.mean(), scale=values.std()), label='Параметрическая оценка')

kde = gaussian\_kde(values)

plt.plot(x, kde(x), label='Усредненная ядерная оценка')

hist\_estimate = hist / (bin\_centers[1] - bin\_centers[0])

plt.plot(bin\_centers, hist\_estimate, label='Гистограммная оценка')

plt.legend()

plt.title('Графики оценок плотности вероятности')

mean\_value = np.mean(values)

median\_value = np.median(values)

variance = np.var(values)

std\_deviation = np.std(values)

print(f"Выборочное среднее: {mean\_value:.4f}")

print(f"Выборочная медиана: {median\_value:.4f}")

print(f"Выборочная дисперсия: {variance:.4f}")

print(f"Выборочное стандартное отклонение: {std\_deviation}")

parametric\_estimate = norm.pdf(bin\_centers, loc=values.mean(), scale=values.std())

kde\_estimate = kde(bin\_centers)

table = PrettyTable()

table.add\_column("№", [i for i in range(1, num\_bins + 1)])

table.add\_column("Центр промежутков", bin\_centers)

table.add\_column("Гистограммная оценка плотности вероятности", hist\_estimate)

table.add\_column("Усредненную ядерную оценку плотности вероятности", kde\_estimate)

table.add\_column("Параметрическая oцeнка плотности вероятности", parametric\_estimate)

print("Значения оценок в средних точках:")

print(table)

table = PrettyTable()

table.add\_column("№", [i for i in range(1, num\_bins + 1)])

table.add\_column("Центр промежутков", bin\_centers)

mse\_parametric = (parametric\_estimate - hist\_estimate) \*\* 2

mse\_kde = (kde\_estimate - hist\_estimate) \*\* 2

table.add\_column("(Усредненная яд. оц. - Гистограммная оц.)^2", mse\_parametric)

table.add\_column("(Параметрическая оц. - Гистограммная оц.)^2", mse\_kde)

table.add\_row(['', '', 'sum', 'sum'])

table.add\_row(['', '', sum(mse\_parametric), sum(mse\_kde)])

print(table)

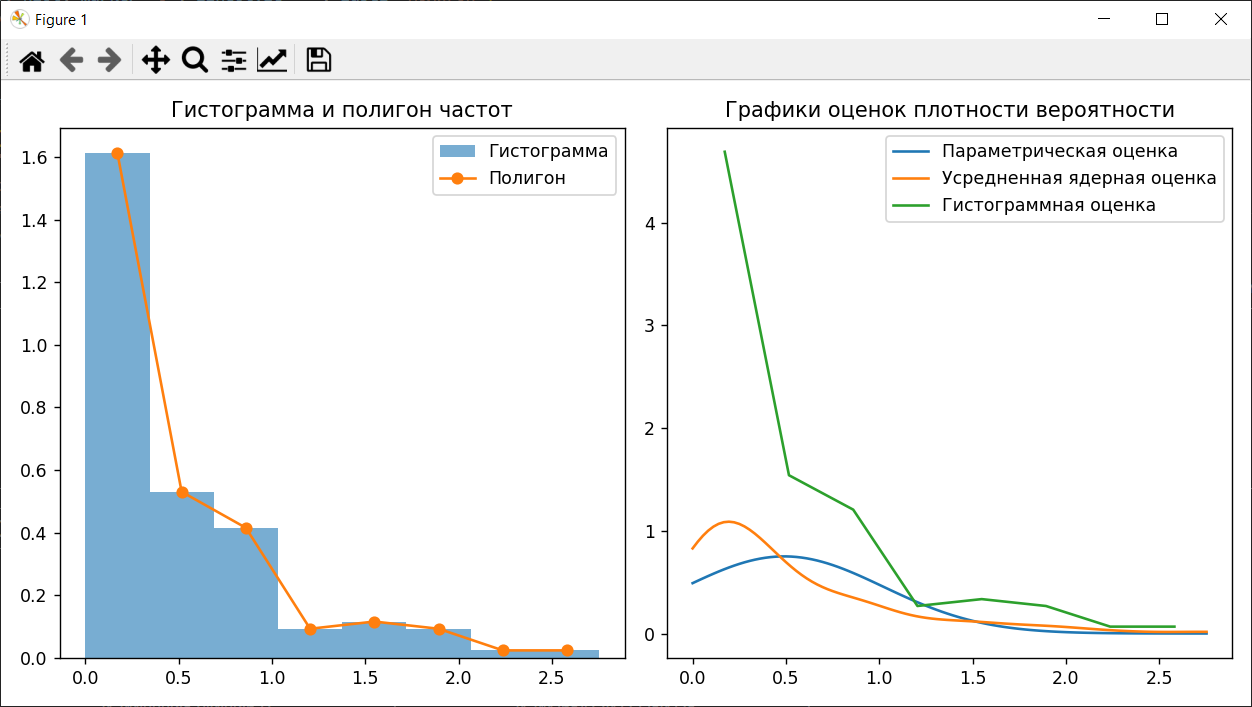
print(f"Среднее квадратичное отклонение параметрической оценки от гистограммной оценки: {np.sqrt(np.mean(mse\_parametric)):.4f}")

print(f"Среднее квадратичное отклонение усредненной ядерной оценки от гистограммной оценки: {np.sqrt(np.mean(mse\_kde)):.4f}")

plt.tight\_layout()

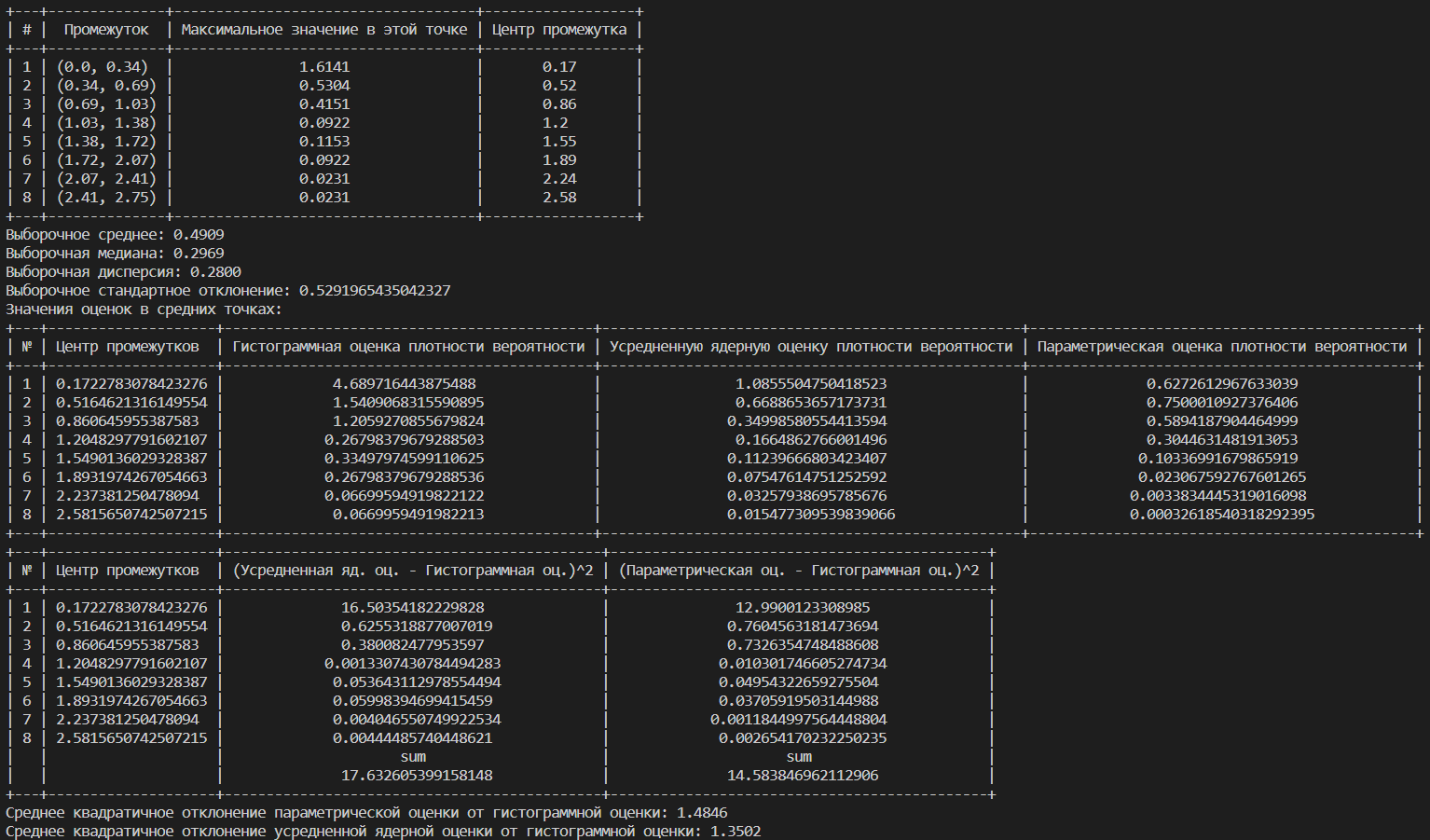
plt.show()

**Результат:**



**Рис. 1.** Графики

Исходя из графиков, можно сделать предположение о геометрическом распределении подборки.



**Рис. 2.** Вычисления

**Задание 3:**

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

1. методом моментов (c помощью указанных в задании моментов);
2. 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 7**

X – выборка из геометрического распределения Gp с параметром p = 0.6. Найти оценку параметра p, считая его неизвестным. Метод моментов реализовать с помощью момента 1-го порядка.

**Листинг:**

import numpy as np

import matplotlib.pyplot as plt

sample\_size = 100

p = 0.6

sample = np.random.geometric(p, size=sample\_size)

p\_estimate\_1 = 1 / np.mean(sample)

print("Оценка параметра p:", p\_estimate\_1)

def likelihood(p, sample):

    likelihood = np.prod(p \* (1-p)\*\*(sample-1))

    return likelihood

grid = np.linspace(0.01, 1, 100)

likelihood\_values = [likelihood(p\_val, sample) for p\_val in grid]

p\_estimate\_2 = grid[np.argmax(likelihood\_values)]

print("Оценка параметра p:", p\_estimate\_2)

plt.plot(grid, likelihood\_values)

plt.axvline(p\_estimate\_1, color='r', linestyle='--', label='Метод моментов')

plt.axvline(p\_estimate\_2, color='g', linestyle=':', label='Метод максимального правдоподобия')

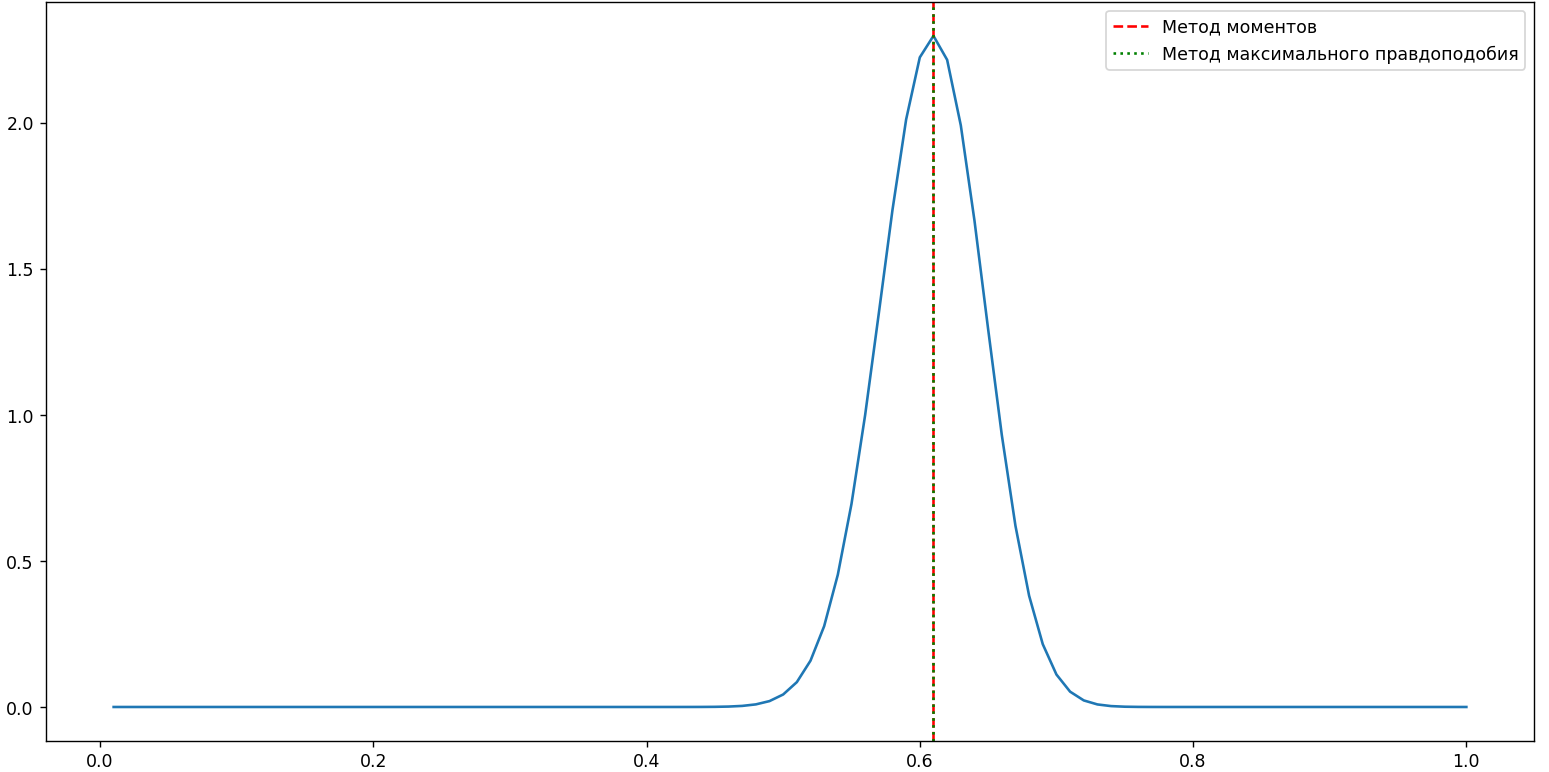
plt.legend()

plt.show()

**Результат:**



**Рис. 1.1.** Результат



**Рис. 1.2.** Результат

**Задание 4:**

Даны две выборки одной случайной величины с нормальным распределением Na,σ2 объема n1 (малый объем, [8; 12]) и n2 (в 70 раз больше n1) соответственно.

**Вариант 7**

1. Для обеих выборок построить точный доверительный интервал уровня доверия q0 для параметра a, считая:
   1. σ неизвестным,
   2. σ известным и равным σ0.
2. В одной системе координат построить графики зависимости длины доверительного интервала от уровня доверия q для всех четырех случаев (объем выборки равен n1, σ неизвестно; объем выборки равен n1, σ известно; объем выборки равен n2, σ неизвестно; объем выборки равен n2, σ известно). При этом q придать минимум 50 разных значений через равные промежутки.

Проанализировать взаимное расположение полученных графиков и объяснить его.

* σ0 = 0.5
* q0 = 0.8

**Листинг:**

***LW4\_1.py***

import numpy as np

from scipy.stats import t, norm

n1 = 10

n2 = 700

q0 = 0.8

a = 1

sigma0 = 0.5

sample1 = np.random.normal(a, sigma0\*\*2, n1)

sample2 = np.random.normal(a, sigma0\*\*2, n2)

CI1 = t.interval(q0, n1-1, loc=np.mean(sample1), scale=np.std(sample1, ddof=1)/np.sqrt(n1))

CI2 = t.interval(q0, n2-1, loc=np.mean(sample2), scale=np.std(sample2, ddof=1)/np.sqrt(n2))

print("Доверительный интервал для выборки 1 (σ неизвестно):", CI1)

print("Доверительный интервал для выборки 2 (σ неизвестно):", CI2)

CI1\_known\_sigma = norm.interval(q0, loc=np.mean(sample1), scale=sigma0/np.sqrt(n1))

CI2\_known\_sigma = norm.interval(q0, loc=np.mean(sample2), scale=sigma0/np.sqrt(n2))

print("Доверительный интервал для выборки 1 (σ известно):", CI1\_known\_sigma)

print("Доверительный интервал для выборки 2 (σ известно):", CI2\_known\_sigma)

***LW4\_2.py***

import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import t, norm

n1 = 10

n2 = 700

sigma0 = 0.5

q\_values = np.linspace(0.01, 0.99, 50)

length\_CI1\_unknown\_sigma = []

length\_CI2\_unknown\_sigma = []

length\_CI1\_known\_sigma = []

length\_CI2\_known\_sigma = []

for q in q\_values:

    CI1\_unknown\_sigma = t.interval(q, n1-1, loc=0, scale=1)

    CI2\_unknown\_sigma = t.interval(q, n2-1, loc=0, scale=1)

    CI1\_known\_sigma = norm.interval(q, loc=0, scale=sigma0 / np.sqrt(n1))

    CI2\_known\_sigma = norm.interval(q, loc=0, scale=sigma0 / np.sqrt(n2))

    length1\_unknown\_sigma = CI1\_unknown\_sigma[1] - CI1\_unknown\_sigma[0]

    length2\_unknown\_sigma = CI2\_unknown\_sigma[1] - CI2\_unknown\_sigma[0]

    length1\_known\_sigma = CI1\_known\_sigma[1] - CI1\_known\_sigma[0]

    length2\_known\_sigma = CI2\_known\_sigma[1] - CI2\_known\_sigma[0]

    length\_CI1\_unknown\_sigma.append(length1\_unknown\_sigma)

    length\_CI2\_unknown\_sigma.append(length2\_unknown\_sigma)

    length\_CI1\_known\_sigma.append(length1\_known\_sigma)

    length\_CI2\_known\_sigma.append(length2\_known\_sigma)

plt.plot(q\_values, length\_CI1\_unknown\_sigma, label="n1, σ unknown")

plt.plot(q\_values, length\_CI2\_unknown\_sigma, label="n2, σ unknown")

plt.plot(q\_values, length\_CI1\_known\_sigma, label="n1, σ known")

plt.plot(q\_values, length\_CI2\_known\_sigma, label="n2, σ known")

plt.xlabel('Уровень доверия, q')

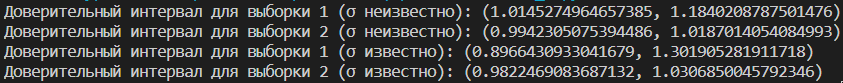
plt.ylabel('Длина доверительного интервала')

plt.title('График зависимости длины доверительного\n интервала от уровня доверия')

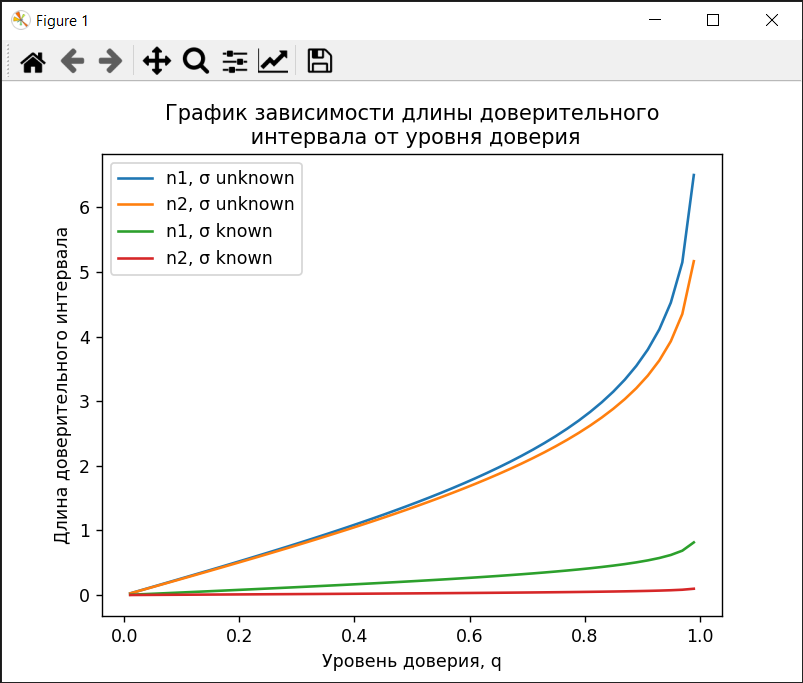
plt.legend()

plt.show()

**Результат:**



**Рис. 1.1.** Результат



**Рис. 1.2.** Результат

Графики для случаев с неизвестной σ имеют более пологий и быстрый рост длины интервала по сравнению с графиками для случаев с известной σ. Для малого объема выборки (n1) доверительный интервал должен быть шире для достижения заданного уровня доверия q. При большом объеме выборки (n2) можно получить более узкий доверительный интервал при заданном уровне доверия q.

**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