|  |  |
| --- | --- |
| **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»** | |

**Практическое занятие №5**

**«Проверка гипотез»**

**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

**Постановка задачи**

Пусть проверяется простая гипотеза относительно параметра распределения : , с заданным уровнем значимости α.

Для нескольких альтернативных гипотез H1: , при . Построить графики мощности критерия значимости, если используется выборка (выборка из ПЗ-2):

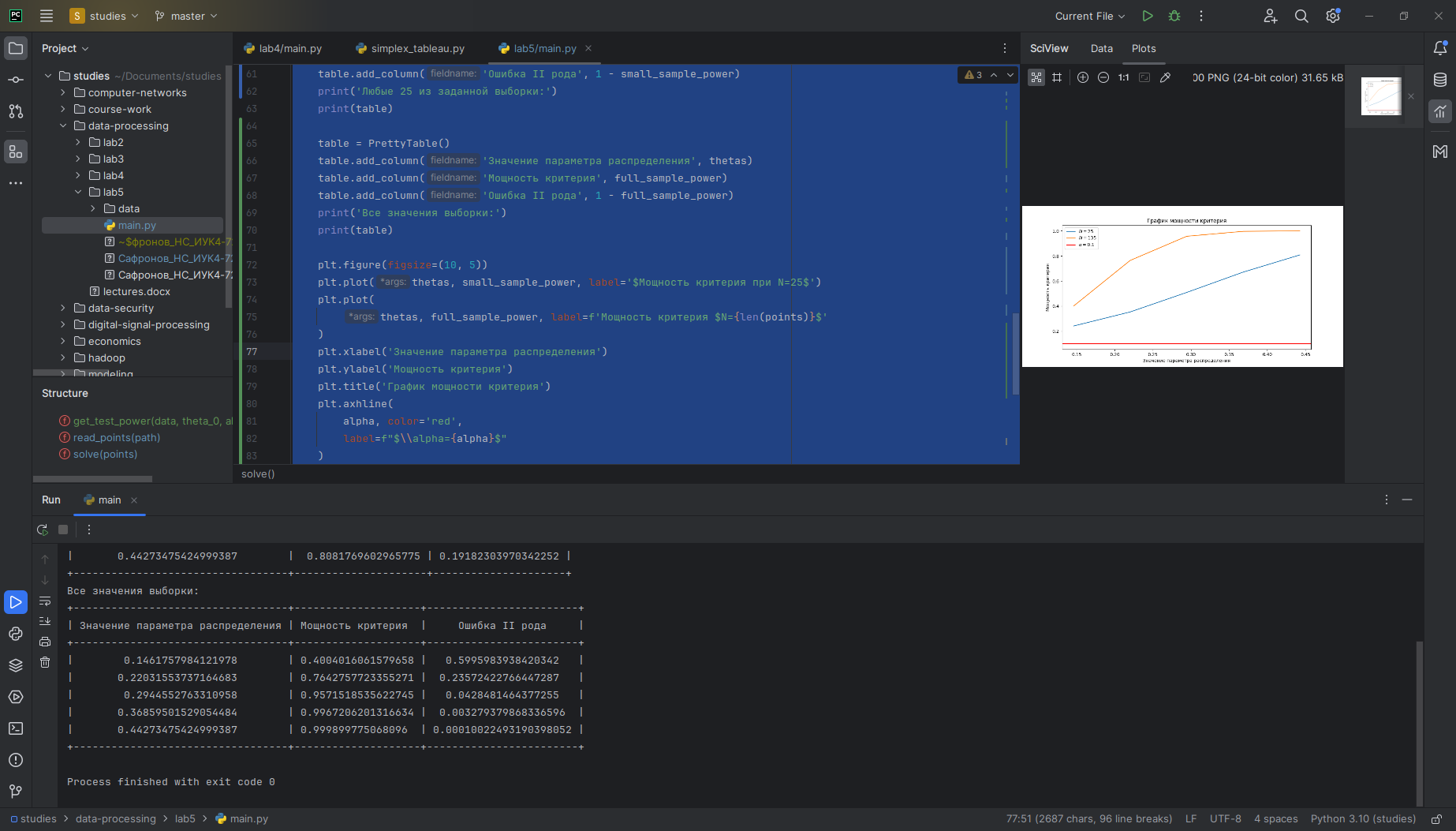
1. объема (любые 25 значений из заданной выборки);

2. объема (полный объем исходной выборки) Используя полученные результаты, построить таблицы "Ошибка II рода и мощность для нескольких альтернативных гипотез с объемом выборки и α" и графики функций мощности критерия для случая 1 и 2.

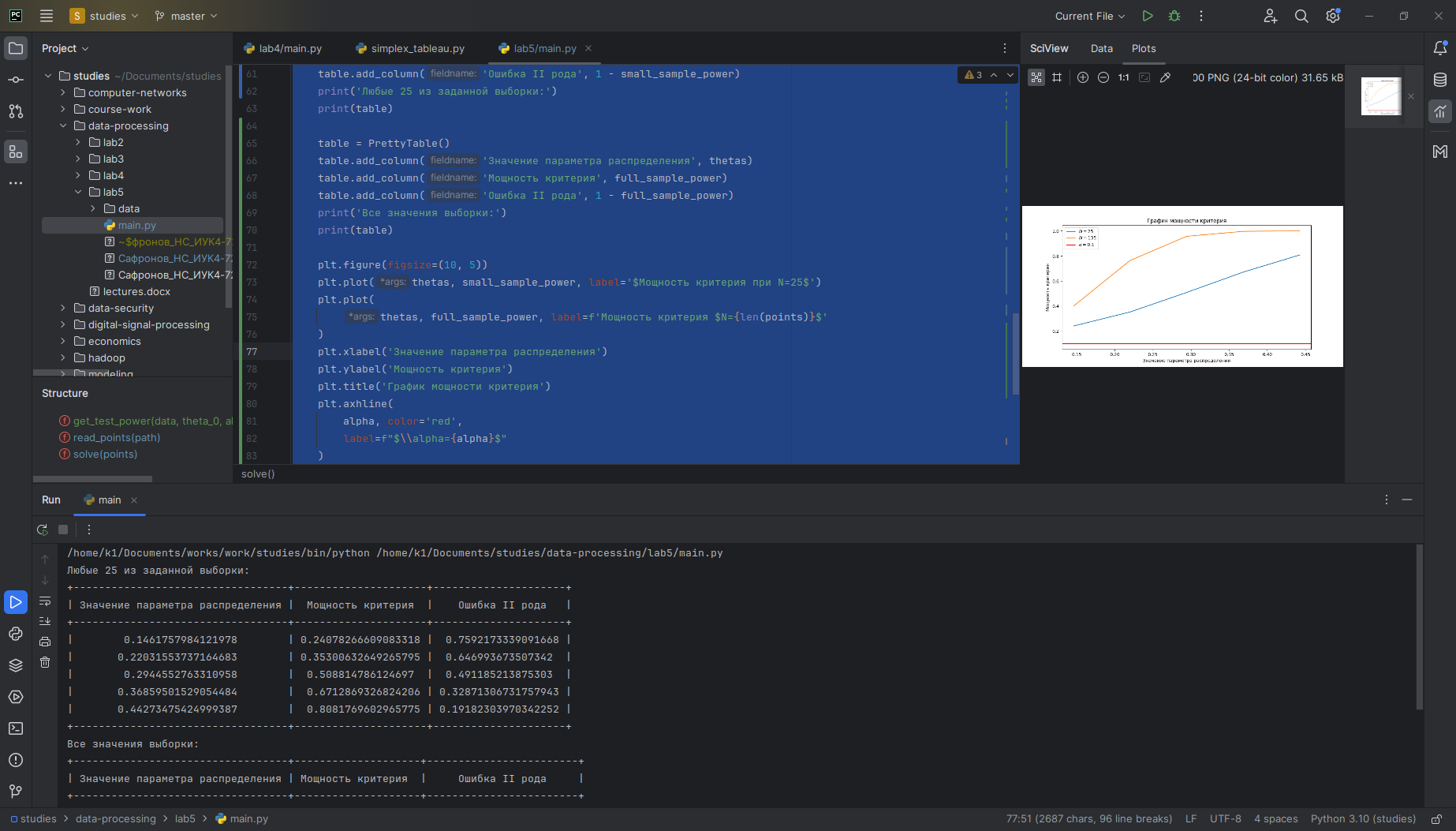
**Ход выполнения практического задания**

За нулевую гипотезу будем брать среднее значение выборки.

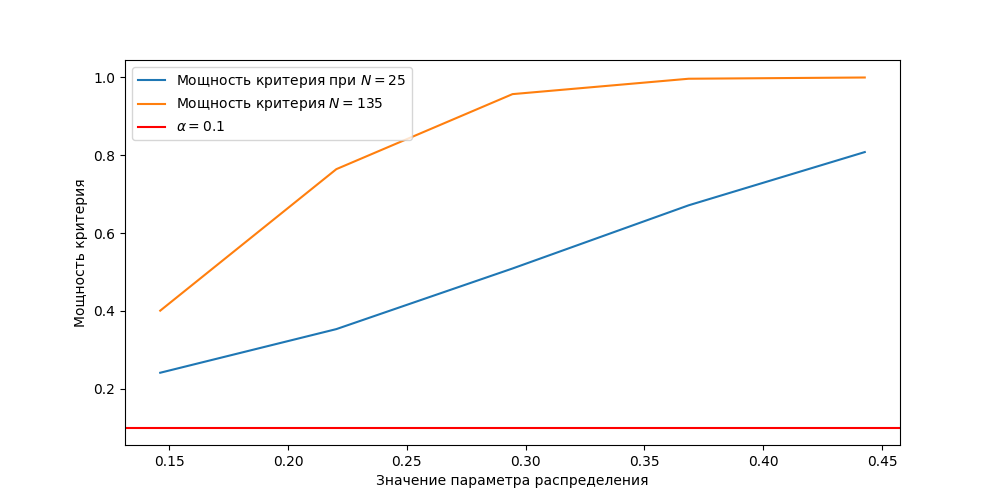
Уровень значимости .

****

**Рисунок 1** – Ошибки II рода и мощность критерия для полной выборки



**Рисунок 3** – Ошибки II рода и мощность критерия для малой выборки ()



**Рисунок 3** – График мощности критерия значимости для малой и полной выборок

**ПРИЛОЖЕНИЯ**

**Листинг программы**

import argparse  
import csv  
  
import matplotlib.pyplot as plt  
import numpy as np  
from prettytable import PrettyTable  
from scipy.stats import norm  
  
  
def get\_test\_power(  
 data: list[float], theta\_0: float, alpha: float, theta: float,  
 sample\_size: int  
) -> float:  
 critical\_value = norm.ppf(1 - alpha)  
 standard\_error = np.std(data) / np.sqrt(sample\_size)  
 critical\_region = (  
 theta\_0 - critical\_value \* standard\_error,  
 theta\_0 + critical\_value \* standard\_error  
 )  
 power = 1 - norm.cdf(critical\_region[1], theta, standard\_error) + norm.cdf(  
 critical\_region[0], theta, standard\_error  
 )  
 return power  
  
  
def read\_points(path: str) -> list[float]:  
 points = []  
  
 with open(path, newline='') as file:  
 reader = csv.reader(file, delimiter=' ', quotechar='|')  
 for row in reader:  
 points.append(float(''.join(row)))  
  
 return points  
  
  
def solve(points: list[float]):  
 alpha = 0.1  
 theta\_0 = np.mean(points)  
 delta = np.std(points) / np.sqrt(len(points))  
 thetas = [theta\_0 + i \* delta for i in range(1, 6)]  
  
 small\_sample\_power = np.array(  
 [  
 get\_test\_power(  
 points, theta\_0, alpha, theta, 25  
 ) for theta in thetas  
 ]  
 )  
 full\_sample\_power = np.array(  
 [  
 get\_test\_power(  
 points, theta\_0, alpha, theta, len(points)  
 ) for theta in thetas  
 ]  
 )  
  
 table = PrettyTable()  
 table.add\_column('Значение параметра распределения', thetas)  
 table.add\_column('Мощность критерия', small\_sample\_power)  
 table.add\_column('Ошибка II рода', 1 - small\_sample\_power)  
 print('Любые 25 из заданной выборки:')  
 print(table)  
  
 table = PrettyTable()  
 table.add\_column('Значение параметра распределения', thetas)  
 table.add\_column('Мощность критерия', full\_sample\_power)  
 table.add\_column('Ошибка II рода', 1 - full\_sample\_power)  
 print('Все значения выборки:')  
 print(table)  
  
 plt.figure(figsize=(10, 5))  
 plt.plot(thetas, small\_sample\_power, label='$Мощность критерия при N=25$')  
 plt.plot(  
 thetas, full\_sample\_power, label=f'Мощность критерия $N={len(points)}$'  
 )  
 plt.xlabel('Значение параметра распределения')  
 plt.ylabel('Мощность критерия')  
 plt.title('График мощности критерия')  
 plt.axhline(  
 alpha, color='red',  
 label=f"$\\alpha={alpha}$"  
 )  
 plt.legend()  
 plt.show()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 parser = argparse.ArgumentParser()  
 parser.add\_argument('-file')  
  
 args = parser.parse\_args()  
 file = args.file or './data/Test14.csv'  
  
 points = read\_points(file)  
 solve(points)