



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 _____ (_____)
(подпись) (Ф.И.О.)

Проверил: _____ (_____)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2023

Постановка задачи

Пусть проверяется простая гипотеза относительно параметра распределения $H_0: \theta = \theta_0$, с заданным уровнем значимости α .

Для нескольких альтернативных гипотез $H_1: \theta = \theta_{1i}$, при $\theta_{1i} = \theta_0 + i\Delta (i = 1, 2, 3, 4, 5)$. Построить графики мощности критерия значимости, если используется выборка (выборка из ПЗ-2):

1. объема $k_1 = 25$ (любые 25 значений из заданной выборки);
2. объема $k_2 = N$ (полный объем исходной выборки) Используя полученные результаты, построить таблицы "Ошибка II рода и мощность для нескольких альтернативных гипотез с объемом выборки k_i 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 $\alpha = 0.1$.

Формула, используемая для вычисления статистики критерия:

$$g = \sqrt{N} * \frac{\bar{x} - a}{\sigma}$$

Значение параметра распределения	Мощность критерия	Ошибка II рода
0.07203605945274881	0.5	0.5
1.3220360594527487	1.0	0.0
2.5720360594527487	1.0	0.0
3.8220360594527487	1.0	0.0
5.072036059452749	1.0	0.0

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

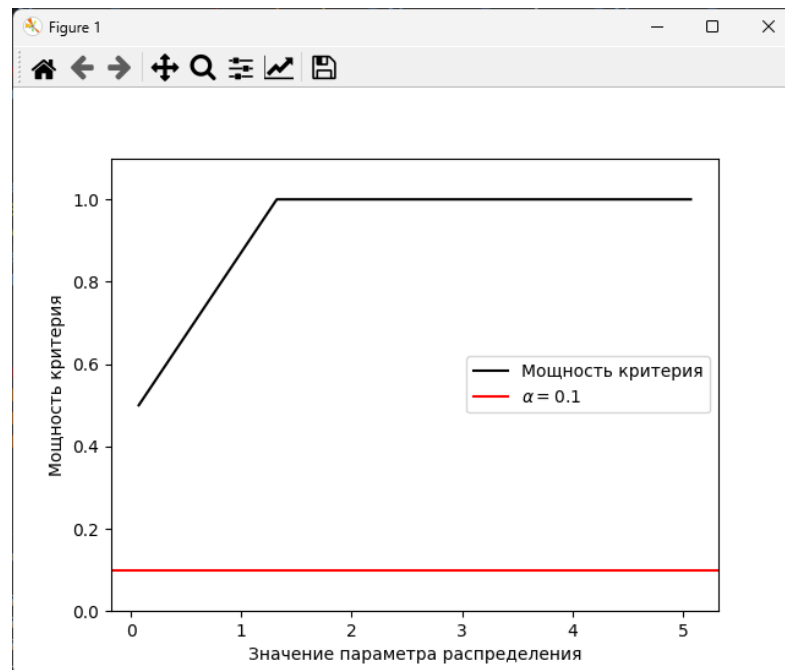


Рисунок 2 – График мощности критерия значимости для полной выборки

Любые 25 значений из заданной выборки:

Значение параметра распределения	Мощность критерия	Ошибка II рода
0.2859853994457233	0.5	0.5
1.5359853994457233	0.99999999592368856	4.0763114439457127e-08
2.7859853994457233	0.9999999999999565	4.3520742565306136e-14
4.035985399445723	1.0	0.0
5.285985399445723	1.0	0.0

Рисунок 3 – Ошибки II рода и мощность критерия для малой выборки (N = 25)

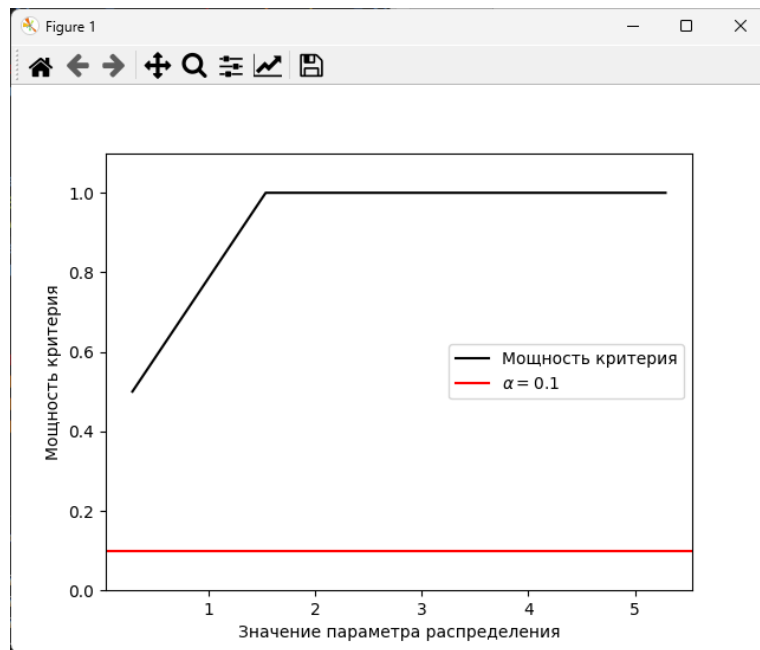


Рисунок 4 – График мощности критерия значимости для малой выборки ($N = 25$)

ПРИЛОЖЕНИЯ

Листинг программы

```
import argparse
import csv
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
from prettytable import PrettyTable

def read_points(path: str) -> list[str]:
    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
    sample_size = len(points)

    sample_mean = np.mean(points)
    sample_std = np.std(points)

    null_hypothesis_mean = sample_mean
    t_statistics = stats.ttest_1samp(points, sample_mean)

    p_value = 2 * (1 - stats.t.cdf(np.abs(t_statistics),
df=sample_size - 1))

    alternative_hypothesis_means =
np.linspace(null_hypothesis_mean, null_hypothesis_mean + 5, 5)

    power_values = np.array([
        1 - stats.t.cdf((null_hypothesis_mean - alt_mean) /
(sample_std / np.sqrt(sample_size))), df=sample_size - 1)
        for alt_mean in alternative_hypothesis_means
    ])

    plt.figure()
    plt.plot(alternative_hypothesis_means, power_values,
label='Мощность критерия')
    plt.xlabel('Значение альтернативной гипотезы')
    plt.ylabel('Мощность критерия')
    plt.title('График мощности критерия')
    plt.axhline(alpha, color='red', label=f'$\\alpha =
{alpha}$')
    plt.ylim(0, 1.1)
```

```

plt.legend()
plt.show()

table = PrettyTable()
table.add_column('Значение параметра распределения',
alternative_hypothesis_means)
table.add_column('Мощность теста', power_values)
table.add_column('Ошибка II рода', 1 - power_values)
print(table)

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)
    print('Полный объём исходных данных:')
    solve(points)

    points = np.random.choice(points, 25)
    print('Любые 25 значений из заданной выборки:')
    solve(points)

```