



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 №4

«Интервальное оценивание»

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

1. Для обеих выборок построить точный доверительный интервал уровня доверия q_0 для параметра σ^2 , считая:

- а) а неизвестным,
- б) а известным и равным a_0 .

2. В одной системе координат построить графики зависимости длины доверительного интервала от уровня доверия q для всех четырех случаев (объем выборки равен n_1 , а неизвестно; объем выборки равен n_1 , а известно; объем выборки равен n_2 , а неизвестно; объем выборки равен n_2 , а известно). При этом q придать минимум 50 разных значений через равные промежутки.

Вариант 14

$$a_0 = 4, q_0 = 0.8$$

Ход выполнения практического задания

Примем размеры малой и большой выборок $n_1 = 15, n_2 = 70 \cdot n_1 = 1050$, соответственно.

Формула доверительного интервала для σ^2 при известном а:

$$P_{a, \sigma^2} \left(\frac{n \cdot s_1^2}{g_2} < \sigma^2 < \frac{n \cdot s_1^2}{g_1} \right) = 1 - \varepsilon, \text{ где}$$

s_1^2 – выборочная дисперсия,

g_1 и g_2 –

квантили распределения χ^2 с n степенями свободы уровня $\alpha = \frac{\varepsilon}{2}$ и $\alpha = 1 - \frac{\varepsilon}{2}$.

Формула доверительного интервала для σ^2 при неизвестном а:

$$P_{a, \sigma^2} \left(\frac{(n-1) \cdot s_0^2}{g_2} < \sigma^2 < \frac{(n-1) \cdot s_0^2}{g_1} \right) = 1 - \varepsilon, \text{ где}$$

s_0^2 – несмещенная выборочная дисперсия,

g_1 и g_2 –

квантили распределения χ^2 с n степенями свободы уровня $\alpha = \frac{\varepsilon}{2}$ и $\alpha = 1 - \frac{\varepsilon}{2}$.

```

Математическое ожидание неизвестно:
Доверительный интервал для  $\sigma^2$  при малой выборке и при уровне доверия 0.8: (0.35, 0.95)
Доверительный интервал для  $\sigma^2$  при большой выборке и при уровне доверия 0.8: (0.88, 0.99)

Математическое ожидание равно  $a_0 = 4$ :
Доверительный интервал для  $\sigma^2$  при малой выборке и при уровне доверия 0.8: (0.33, 0.87)
Доверительный интервал для  $\sigma^2$  при малой выборке и при уровне доверия 0.8: (0.88, 0.99)

```

Рисунок 1 – Полученные доверительные интервалы

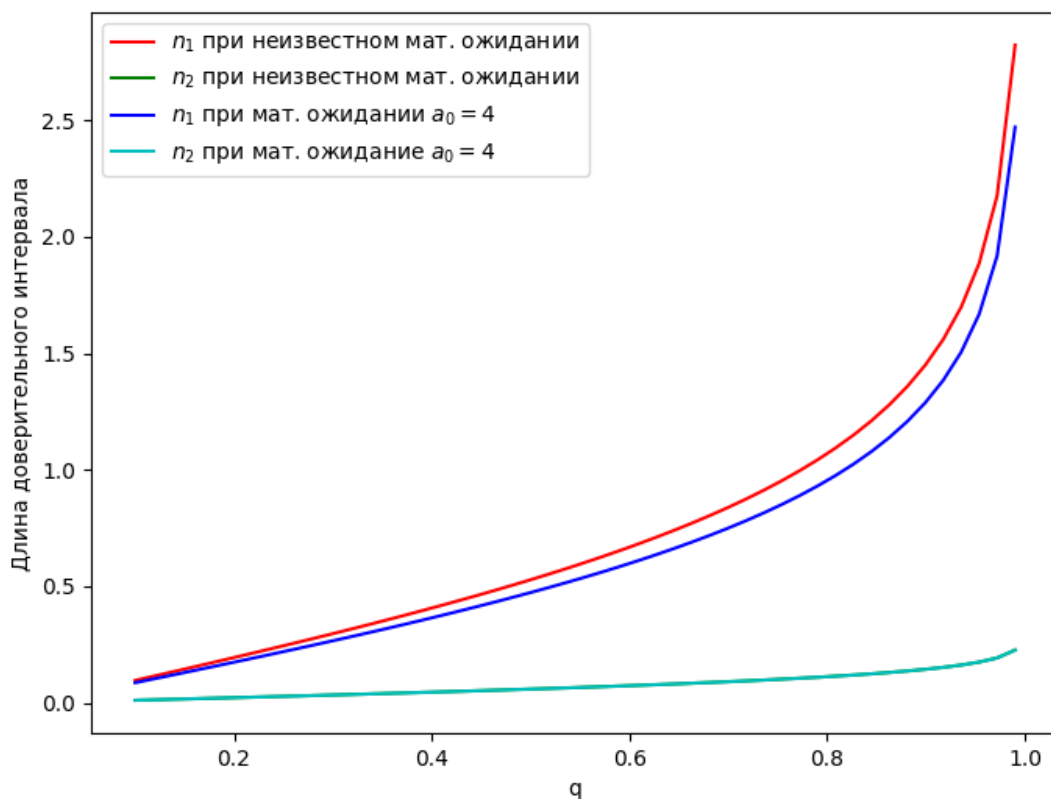


Рисунок 2 - Графики зависимости длины доверительного интервала от уровня доверия q

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.

ПРИЛОЖЕНИЯ

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  $\sigma^2$  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  $\sigma^2$  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  $\sigma^2$  при малой выборке и при уровне "
        f"доверия {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  $\sigma^2$  при малой выборке и при уровне "
        f"доверия {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()

```