

```
import pandas as pd
import numpy as np
import scipy.stats as st
```

Лабораторная работа №2

Задание 3

{Фамилия Имя}, {Номер группы}, Вариант {Номер варианта}, ({Дата})

По данным независимых равноточных измерений физической величины, a оценить истинное значение измеряемой величины и точность измерений с надежностью γ .

Данные

```
LB_NUM = 2
LB_SURNAME = 'Фамилия'
LB_NAME = "Имя"
LB_VARIANT = 23
LB_EXERCISE_NUM = 3
print(f'Фамилия, Имя: {LB_SURNAME}, {LB_NAME}\nНомер лабораторной работы: {LB_NUM}\nВариант лабораторной работы: {LB_VARIANT}\nНомер задания лабораторной работы: {LB_EXERCISE_NUM}')
```

Фамилия, Имя: Фамилия, Имя
Номер лабораторной работы: 2
Вариант лабораторной работы: 23
Номер задания лабораторной работы: 3

```
data = pd.Series([50.0, 43.7, 35.2, 45.6, 39.8, 37.2, 36.7, 46.2, 35.9, 36.4, 36.7, 48.5, 39.6, 40.6, 45.2, 48.4])
confidence_prob = .95
data
```

0	50.0
1	43.7
2	35.2
3	45.6

```
4      39.8
5      37.2
6      36.7
7      46.2
8      35.9
9      36.4
10     36.7
11     48.5
12     39.6
13     40.6
14     45.2
15     48.4
dtype: float64
```

```
data.describe()
```

```
count      16.000000
mean       41.606250
std         5.131922
min        35.200000
25%        36.700000
50%        40.200000
75%        45.750000
max        50.000000
dtype: float64
```

Всего элементов ряда

```
data_len = len(data)
data_len
```

```
16
```

Выборочное среднее

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

```
mean = np.mean(data)
mean
```

41.60625

Выборочная дисперсия

$$s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

```
var = np.var(data, ddof=1)
var
```

26.336624999999994

Среднеквадратическое отклонение

$$s = \sqrt{s^2}$$

```
std = np.std(data, ddof=1)
std
```

5.131922154514816

Доверительный интервал для математического ожидания при неизвестной генеральной дисперсии σ

$$\left(\bar{x} - t(\gamma, n-1) \cdot \frac{s}{\sqrt{n}}, \quad \bar{x} + t(\gamma, n-1) \cdot \frac{s}{\sqrt{n}} \right)$$

Считаем интервал без модуля stats

```
def confidence_interval_for_expectation_t(data, confidence_prob):
    mean = np.mean(data)
    std = np.std(data, ddof=1)
    n = len(data)
    # t.ppf считает левостороннюю область, а нам требуется двусторонняя
    alpha = (1 - confidence_prob) / 2
    gamma = 1 - alpha
    t_value = st.t.ppf(gamma, n - 1)
    delta = t_value * std / np.sqrt(n)
    return (mean - delta, mean + delta)

confidence_interval_for_expectation_t(data, confidence_prob)
```

```
(38.8716417139784, 44.34085828602161)
```

Считаем интервал с модулем stats

```
interval_expectation = st.t.interval(confidence_prob, df=data_len - 1,  
loc=mean, scale=st.sem(data))  
interval_expectation
```

```
(38.8716417139784, 44.34085828602161)
```

```
print(f"Вывод: Таким образом с вероятностью {confidence_prob} можно  
утверждать, \nчто математическое ожидание лежит в интервале  
{interval_expectation}")
```

```
Вывод: Таким образом с вероятностью 0.95 можно утверждать,  
что математическое ожидание лежит в интервале (38.8716417139784,  
44.34085828602161)
```