

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

Лабораторная работа №2

Задание 1

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

По данным выборки, удовлетворяющей нормальному закону распределения, вычислить:

1. выборочное среднее;
2. исправленное выборочное среднее квадратическое отклонение;
3. доверительный интервал для математического ожидания при доверительной вероятности γ ;
4. доверительный интервал для среднего квадратического отклонения для того же значения γ .

Данные

```
LB_NUM = 2
LB_SURNAME = 'Фамилия'
LB_NAME = "Имя"
LB_VARIANT = 23
LB_EXERCISE_NUM = 1
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: 1

```
all_data = pd.read_json('input/Данные к заданию №1.json')
variant_data = all_data[all_data["Вариант"] == LB_VARIANT]
```

```
data = pd.Series(variant_data["Данные"].iloc[0], dtype=float)
confidence_prob = variant_data["Доверительная вероятность"].iloc[0]
```

```
print(f"Доверительная вероятность: {confidence_prob}")
```

Доверительная вероятность: 0.999

data

```
0    35.5
1    38.4
2    35.4
3    31.4
4    29.3
5    38.3
6    27.9
7    35.6
8    35.6
9    36.7
10   31.4
11   34.7
12   34.9
13   27.7
14   25.6
15   32.5
16   31.7
17   31.8
dtype: float64
```

```
data.describe()
```

```
count    18.000000
mean     33.022222
std       3.710170
min      25.600000
25%      31.400000
50%      33.600000
75%      35.575000
```

```
max      38.400000
dtype: float64
```

Всего элементов ряда

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

```
18
```

Выборочное среднее

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

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

```
33.02222222222222
```

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
```

```
13.765359477124184
```

Среднеквадратическое отклонение

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

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

```
3.7101697369694806
```

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)
```

```
(29.55473757023023, 36.48970687421421)
```

Считаем интервал с модулем stats

```
st.sem(data) == std/np.sqrt(data_len)
```

```
True
```

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

```
(29.554737570230277, 36.48970687421421)
```

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

Вывод: Таким образом с вероятностью 0.999 можно утверждать,
что математическое ожидание лежит в интервале (29.554737570230277,

36.48970687421421)

Доверительный интервал для среднего квадратического отклонения для того же значения γ

Через χ^2

$$\left(\frac{S\sqrt{n-1}}{\chi_{\text{left}}}, \frac{S\sqrt{n-1}}{\chi_{\text{right}}} \right)$$

```
def confidence_interval_std(data, confidence_prob):
    std = np.std(data, ddof=1)
    n = len(data)
    chi_left = np.sqrt(st.chi2.ppf((1 + confidence_prob) / 2, n - 1))
    chi_right = np.sqrt(st.chi2.ppf((1 - confidence_prob) / 2, n - 1))
    return (std * np.sqrt(n - 1) / chi_left, std * np.sqrt(n - 1) /
chi_right)

confidence_interval_std(data, confidence_prob)
```

(2.3361183278849347, 7.66773403328059)

```
print(f"Вывод: Таким образом с вероятностью {confidence_prob} можно
утверждать, что стандартное отклонение генеральной\нсовокупности лежит в
интервале {confidence_interval_std(data, confidence_prob)}")
```

Вывод: Таким образом с вероятностью 0.999 можно утверждать, что стандартное отклонение генеральной совокупности лежит в интервале (2.3361183278849347, 7.66773403328059)