

Задание 1.

Исходные данные:

Известно, что генеральная совокупность распределена нормально со средним квадратическим отклонением, равным 16.

Найти доверительный интервал для оценки математического ожидания μ с надежностью 0.95, если выборочная средняя $M = 80$, а объем выборки $n = 256$.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
```

```
>>> a = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 44, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156])
```

```
>>> a.mean()
```

```
80.0625
```

```
>>>
```

```
>>> a.std(ddof=1)
```

```
57.190942653261345
```

```
>>>
```

```
>>> a.shape
```

```
(256,)
```

```
>>>
```

```
>>> a.std(ddof=1) / np.sqrt(a.shape[0])
```

```
3.574433915828834
```

```
>>>
```

```
>>> def statistic(a: np.ndarray) -> float:
```

```
...     return (a.mean() - 1) / (0.95 / np.sqrt(a.shape[0]))
```

...

```
>>> alpha = 0.95
>>> import scipy
>>> from scipy import stats
>>> t1 = stats.norm.ppf(alpha / 2)
>>> t2 = stats.norm.ppf(1 - alpha / 2)
>>> print(t1, t2)
-0.06270677794321385 0.06270677794321385
>>>
>>> s = statistic(a)
>>> print(s)
1331.578947368421
>>>
>>>
```

Задание 2.

Исходные данные:

В результате 10 независимых измерений некоторой величины X , выполненных с одинаковой точностью, получены опытные данные:

6.9, 6.1, 6.2, 6.8, 7.5, 6.3, 6.4, 6.9, 6.7, 6.1

Предполагая, что результаты измерений подчинены нормальному закону распределения вероятностей, оценить истинное значение величины X при помощи доверительного интервала, покрывающего это значение с доверительной вероятностью 0,95.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
>>> a = np.array([6.9, 6.1, 6.2, 6.8, 7.5, 6.3, 6.4, 6.9, 6.7, 6.1])
>>> a.mean()
6.5900000000000001
```

```

>>>
>>> a.std(ddof=1)
0.4508017549014448
>>>
>>> a.shape
(10,)
>>>
>>> a.std(ddof=1) / np.sqrt(a.shape[0])
0.142556031868954
>>>
>>> def statistic(a: np.ndarray) -> float:
...     return (a.mean() - 1) / (0.95 / np.sqrt(a.shape[0]))
...
>>> alpha = 0.95
>>> import scipy
>>> from scipy import stats
>>> t1 = stats.norm.ppf(alpha / 2)
>>> t2 = stats.norm.ppf(1 - alpha / 2)
>>> print(t1, t2)
-0.06270677794321385 0.06270677794321385
>>>
>>> s = statistic(a)
>>> print(s)
18.607507495096048
>>>
>>>

```

Задание 3.

Исходные данные:

Утверждается, что шарики для подшипников, изготовленные автоматическим станком, имеют средний диаметр 17 мм.

Используя односторонний критерий с $\alpha=0,05$, проверить эту гипотезу, если в выборке из $n=100$ шариков средний диаметр

оказался равным 17.5 мм, а дисперсия известна и равна 4 мм.

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```
>>> import numpy as np
```

```
>>> a = np.array([17.0, 17.1, 16.9, 17.0, 17.2, 16.8, 16.5, 17.0, 17.6, 16.0, 17.8, 18.0, 17.0, 17.5, 17.3, 17.7, 17.4, 17.7, 17.9, 17.0, 17.3, 17.8, 17.9, 17.4, 17.6, 17.0, 17.2, 18.0, 17.1, 17.8, 17.9, 17.5, 17.2, 17.5, 17.8, 17.3, 17.6, 17.9, 17.0, 17.8, 16.8, 17.0, 17.7, 16.7, 17.9, 17.6, 17.5, 17.8, 17.5, 17.9, 17.2, 17.8, 17.9, 17.4, 17.8, 17.2, 17.5, 17.9, 17.7, 17.5, 17.9, 17.8, 17.2, 17.7, 17.9, 17.9, 17.8, 17.9, 17.1, 17.3, 17.9, 17.5, 17.8, 17.9, 17.3, 17.8, 17.6, 17.5, 17.6, 17.8, 17.4, 17.8, 17.2, 17.5, 17.9, 17.2, 17.9, 17.8, 17.2, 17.9, 17.8, 17.5, 17.4, 17.2, 17.5, 17.9, 17.4, 17.6, 17.5, 17.9])
```

```
>>> a = np.array([17.0, 17.1, 16.9, 17.0, 17.2, 16.8, 16.5, 17.0, 17.6, 16.0, 17.8, 18.0, 17.0, 17.5, 17.3, 17.7, 17.4, 17.7, 17.9, 17.0, 17.3, 17.8, 17.9, 17.4, 17.6, 17.0, 17.2, 18.0, 17.1, 17.8, 17.9, 17.5, 17.2, 17.5, 17.8, 17.3, 17.6, 17.9, 17.0, 17.8, 16.8, 17.0, 17.7, 16.7, 17.9, 17.6, 17.5, 17.8, 17.5, 17.9, 17.2, 17.8, 17.9, 17.4, 17.8, 17.2, 17.5, 17.9, 17.7, 17.5, 17.9, 17.8, 17.2, 17.7, 17.9, 17.9, 17.8, 17.9, 17.1, 17.3, 17.9, 17.5, 17.8, 17.9, 17.3, 17.8, 17.6, 17.5, 17.6, 17.8, 17.4, 17.8, 17.2, 17.5, 17.9, 17.2, 17.9, 17.8, 17.2, 17.9, 17.8, 17.5, 17.4, 17.2, 17.5, 17.9, 17.4, 17.6, 17.5, 17.9])
```

```
>>> a.mean()
```

```
17.5
```

```
>>> a.std(ddof=1)
```

```
0.3776589236682518
```

```
>>>
```

```
>>> a.shape
```

```
(100,)
```

```
>>>
```

```
>>> a.std(ddof=1) / np.sqrt(a.shape[0])
```

```
0.037765892366825184
```

```
>>>
```

```
>>> def statistic(a: np.ndarray) -> float:
```

```
...     return (a.mean() - 1) / (0.05 / np.sqrt(a.shape[0]))
```

```
...
```

```

>>> alpha = 0.05
>>> import scipy
>>> from scipy import stats
>>> t1 = stats.norm.ppf(alpha / 0.5)
>>> t2 = stats.norm.ppf(1 - alpha / 0.5)
>>> print(t1, t2)
-1.2815515655446004 1.2815515655446004
>>>
>>> s = statistic(a)
>>> print(s)
3300.0
>>>

```

Задание 4.

Исходные данные:

Продавец утверждает, что средний вес пачки печенья составляет 200 г.

Из партии извлечена выборка из 10 пачек. Вес каждой пачки составляет:

202, 203, 199, 197, 195, 201, 200, 204, 194, 190.

Известно, что их веса распределены нормально.

Верно ли утверждение продавца, если учитывать, что доверительная вероятность равна 99%?

Решение:

Python 3.8.10 (default, Jun 2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

```

>>> import numpy as np
>>> a = np.array([202, 203, 199, 197, 195, 201, 200, 204, 194, 190])
>>> a.mean()
198.5
>>> a.std(ddof=1)
4.453463071962462
>>>
>>> a.shape

```

```

(10,)
>>>
>>> a.std(ddof=1) / np.sqrt(a.shape[0])
1.4083086782851737
>>>
>>> def statistic(a: np.ndarray) -> float:
...     return (a.mean() - 1) / (0.99 / np.sqrt(a.shape[0]))
...
>>> alpha = 0.99
>>> import scipy
>>> from scipy import stats
>>> t1 = stats.norm.ppf(alpha / 2)
>>> t2 = stats.norm.ppf(1 - alpha / 2)
>>> print(t1, t2)
-0.012533469508069276 0.012533469508069276
>>>
>>> s = statistic(a)
>>> print(s)
630.8584221042979
>>>
Да верное.

```