

# Теория вероятностей и математическая статистика

## Задание 1.

Известно, что генеральная совокупность распределена нормально со средним квадратическим отклонением, равным 16. Найти доверительный интервал для оценки математического ожидания  $\mu$  с надежностью 0.95, если выборочная средняя  $M = 80$ , а объем выборки  $n = 256$ .

### Подсказка № 1

Поскольку стандартное отклонение генеральной совокупности известно, и распределение нормально, используйте Z-распределение для нахождения критического значения. Используйте функцию `scipy.stats.norm.ppf()` для вычисления критического значения Z для выбранного уровня значимости.

### Подсказка № 2

Стандартная ошибка определяется как  $\sigma/\sqrt{n}$ , где  $\sigma$  — стандартное отклонение генеральной совокупности, а  $n$  — размер выборки. Убедитесь, что правильно применили эту формулу в Python.

### Подсказка № 3

Доверительный интервал для среднего значения рассчитывается как выборочная средняя  $\pm$  (критическое значение  $Z \times$  стандартная ошибка). Это можно реализовать в Python, вычислив погрешность интервала и затем добавив и вычтя её из выборочной средней.

### Подсказка № 4

Используйте функции из библиотеки `scipy.stats` для нахождения критических значений и стандартного нормального распределения. Функция `ppf` помогает найти критическое значение Z для заданного уровня значимости, а `np.sqrt` для вычисления квадратного корня.

### Эталонное решение:

```
import scipy.stats as stats
import numpy as np
```

```

# Данные задачи

sigma = 16 # Стандартное отклонение генеральной совокупности

M = 80 # Выборочная средняя

n = 256 # Объем выборки

alpha = 0.05 # Уровень значимости для 95% доверительного интервала


# Критическое значение для Z-распределения

z_crit = stats.norm.ppf(1 - alpha / 2)


# Расчет стандартной ошибки

standard_error = sigma / np.sqrt(n)


# Доверительный интервал

margin_of_error = z_crit * standard_error

confidence_interval = (M - margin_of_error, M + margin_of_error)


print(f"Доверительный интервал: {confidence_interval}")

```

## Задача 2.

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.

### Подсказка № 1

Перед вычислением доверительного интервала сначала нужно найти выборочную среднюю и стандартное отклонение. В Python используйте функции `np.mean()` и `np.std()` (с параметром `ddof=1` для стандартного отклонения выборки).

## Подсказка № 2

Для построения доверительного интервала с использованием  $t$ -распределения, найдите критическое значение  $t$ , соответствующее вашему уровню значимости ( $\alpha$ ) и числу степеней свободы. В Python это можно сделать с помощью функции `stats.t.ppf()`, где `df` — это число степеней свободы ( $n - 1$  для одностороннего теста).

## Подсказка № 3

Стандартная ошибка выбирается как стандартное отклонение выборки, деленное на квадратный корень из размера выборки. Это можно рассчитать как `std_dev_X / np.sqrt(n)`.

## Подсказка № 4

Доверительный интервал рассчитывается как выборочная средняя  $\pm$  (критическое значение  $t \times$  стандартная ошибка). В Python это можно реализовать простым умножением и сложением/вычитанием.

## Эталонное решение:

```
import numpy as np

import scipy.stats as stats

# Данные задачи

data = [6.9, 6.1, 6.2, 6.8, 7.5, 6.3, 6.4, 6.9, 6.7, 6.1]

n = len(data) # Объем выборки

alpha = 0.05 # Уровень значимости для 95% доверительного интервала

# Среднее значение и стандартное отклонение выборки

mean_X = np.mean(data)

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

# Критическое значение для t-распределения

t_crit = stats.t.ppf(1 - alpha / 2, df=n - 1)
```

```

# Расчет стандартной ошибки

standard_error = std_dev_X / np.sqrt(n)

# Доверительный интервал

margin_of_error = t_crit * standard_error

confidence_interval = (mean_X - margin_of_error, mean_X +
margin_of_error)

print(f"Доверительный интервал: {confidence_interval}")

```

### Задача 3.

Рост дочерей 175, 167, 154, 174, 178, 148, 160, 167, 169, 170 Рост матерей 178, 165, 165, 173, 168, 155, 160, 164, 178, 175 Используя эти данные построить 95% доверительный интервал для разности среднего роста родителей и детей.

#### Подсказка № 1

Начните с разделения данных на две группы: рост матерей и рост дочерей. Для каждой группы данных рассчитайте выборочное среднее и дисперсию. Используйте функции `np.mean()` для среднего и `np.var()` с параметром `ddof=1` для дисперсии выборки. Это важно для точности последующих расчетов.

#### Подсказка № 2

Для двух независимых выборок необходимо рассчитать объединённую дисперсию (пуленую дисперсию). Она рассчитывается как взвешенная средняя дисперсий обеих групп, с учетом их объемов. Формула:

$$\text{pooled\_variance} = \frac{(n1-1)\text{var\_daughters} + (n2-1)\text{var\_mothers}}{n1+n2-2}$$

#### Подсказка № 3

Для двустороннего теста с 95% доверительным интервалом найдите критическое значение t из t-распределения. В Python используйте `stats.t.ppf()` с параметром `df` равным  $n1+n2-2$  и уровнем значимости  $\alpha=0.05$ .

#### Подсказка № 4

Стандартная ошибка разности средних рассчитывается как корень из суммы долей объединённой дисперсии, деленной на объёмы выборок. Формула:

$$\text{standard\_error\_diff} = \sqrt{\text{pooled\_variance} * \left( \frac{1}{n_1} + \frac{1}{n_2} \right)}$$

### Подсказка № 5

Используйте стандартную ошибку разности средних и критическое значение  $t$  для вычисления границ доверительного интервала. Доверительный интервал рассчитывается как разность средних значений минус и плюс произведение критического значения  $t$  на стандартную ошибку разности.

### Эталонное решение:

```
import numpy as np

import scipy.stats as stats

# Данные задачи

mothers_height = [178, 165, 165, 173, 168, 155, 160, 164, 178, 175]

daughters_height = [175, 167, 154, 174, 178, 148, 160, 167, 169, 170]

# Объем выборок

n1 = len(mothers_height)

n2 = len(daughters_height)

# Средние значения и дисперсии

mean_mothers = np.mean(mothers_height)

mean_daughters = np.mean(daughters_height)

var_mothers = np.var(mothers_height, ddof=1)

var_daughters = np.var(daughters_height, ddof=1)

# Объединённая дисперсия
```

```
pooled_variance = ((n1 - 1) * var_mothers + (n2 - 1) *
var_daughters) / (n1 + n2 - 2)

# Критическое значение для t-распределения
alpha = 0.05

t_crit = stats.t.ppf(1 - alpha / 2, df=n1 + n2 - 2)

# Расчет стандартной ошибки разности средних
standard_error_diff = np.sqrt(pooled_variance * (1 / n1 + 1 / n2))

# Доверительный интервал
margin_of_error = t_crit * standard_error_diff

confidence_interval = (mean_mothers - mean_daughters -
margin_of_error,

                        mean_mothers - mean_daughters +
margin_of_error)

print(f"Доверительный интервал для разности средних:
{confidence_interval}")
```