



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Дальневосточный федеральный университет»
(ДФУ)**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ**

**Лабораторная работа №5
Тестирование гипотезы о среднем**

Дисциплина «Теория вероятностей и математическая статистика»

Студент группы Б9123-01.03.02ии
Моттуева Уруйдана Михайловна

г. Владивосток

2025

Целью данной работы является изучение и реализация z-теста и t-теста для среднего нормальной генеральной совокупности.

Z-тест

Используется, когда известно стандартное отклонение генеральной совокупности σ .

$$Z = \frac{(\bar{X} - \mu_0)\sqrt{n}}{\sigma} \sim N(0; 1).$$

где:

- \bar{X} – выборочное среднее,
- μ_0 – гипотетическое среднее,
- σ – стандартное отклонение генеральной совокупности,
- n – размер выборки.

P-значение:

$$p = \begin{cases} 1 - \Phi(z), & H_1: \mu > \mu_0 \\ \Phi(z), & H_1: \mu < \mu_0 \\ 2\Phi(-|z|), & H_1: \mu \neq \mu_0 \end{cases}$$

Где $\Phi(x)$ это

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt.$$

Реализация

```
import numpy as np
from scipy.stats import norm, t

def mean_hypothesis_test(sample, mu0, alternative='two-sided', std=None):
    n = len(sample)
    sample_mean = np.mean(sample)

    if std is not None: # Z-тест (известно sigma)
        z = (sample_mean - mu0) / (std / np.sqrt(n))
        if alternative == 'two-sided':
            p_value = 2 * (1 - norm.cdf(abs(z)))
        elif alternative == 'less':
            p_value = norm.cdf(z)
        elif alternative == 'greater':
            p_value = 1 - norm.cdf(z)
        else:
            raise ValueError("alternative must be 'two-sided', 'less', or 'greater'")
    return p_value
```

Т-тест

Используется, когда стандартное отклонение неизвестно (оценивается по выборке).

$$T_{n-1} = \frac{(\bar{X} - \mu_0)\sqrt{n}}{\hat{\sigma}_X^2} \sim T(n-1)$$

где $\hat{\sigma}_X^2$ – несмещенная выборочная дисперсия.

P-значение:

$$p = \begin{cases} 1 - F_{T_{n-1}}(t), & H_1: \mu > \mu_0 \\ F_{T_{n-1}}(t), & H_1: \mu < \mu_0, \\ 2F_{T_{n-1}}(-|t|), & H_1: \mu \neq \mu_0 \end{cases}$$

где $F_{T_{n-1}}$ – функция распределения Стьюдента с $n - 1$ степенью свободы.

Плотность распределения Стьюдента:

$$f_{T_k}(t) = \frac{\Gamma\left(\frac{k+1}{2}\right)}{\sqrt{(\pi k)}\Gamma\left(\frac{k}{2}\right)} \left(1 + \frac{t^2}{k}\right)^{-\frac{k+1}{2}},$$
$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$$

Реализация без scipy

```
    return p_value
else: # t-тест (sigma неизвестно)
    s = np.std(sample, ddof=1)
    t_stat = (sample_mean - mu0) / (s / np.sqrt(n))
    if alternative == 'two-sided':
        p_value = 2 * (1 - t.cdf(abs(t_stat), df=n-1))
    elif alternative == 'less':
        p_value = t.cdf(t_stat, df=n-1)
    elif alternative == 'greater':
        p_value = 1 - t.cdf(t_stat, df=n-1)
    else:
        raise ValueError("alternative must be 'two-sided', 'less', or 'greater'")
    return p_value
```

Реализация scipy

```
from scipy.stats import ttest_1samp

t_stat, p_value_scipy = ttest_1samp(sample, popmean=mu0)
```

Задача 1

в) Установлено, что средний вес таблетки лекарства сильного действия должен быть равен $a_0 = 0,50$ мг. Выборочная проверка 121 таблетки полученной партии лекарства показала, что средний вес таблетки этой партии $\bar{x} = 0,53$ мг. Требуется при уровне значимости 0,01 проверить нулевую гипотезу $H_0: a = a_0 = 0,50$ при конкурирующей гипотезе $H_1: a > 0,50$. Многократными предварительными опытами по взвешиванию таблеток, поставляемых фармацевтическим заводом, было установлено, что вес таблеток распределен нормально со средним квадратическим отклонением $\sigma = 0,11$ мг.

```
[13] import numpy as np
      from scipy.stats import norm, t

      def mean_hypothesis_test(sample, mu0, alternative='two-sided', std=None):
          n = len(sample)
          sample_mean = np.mean(sample)

          if std is not None: # Z-тест (известно sigma)
              z = (sample_mean - mu0) / (std / np.sqrt(n))
              if alternative == 'two-sided':
                  p_value = 2 * (1 - norm.cdf(abs(z)))
              elif alternative == 'less':
                  p_value = norm.cdf(z)
              elif alternative == 'greater':
                  p_value = 1 - norm.cdf(z)
              else:
                  raise ValueError("alternative must be 'two-sided', 'less', or 'greater'")
              return p_value
          else: # t-тест (sigma неизвестно)
              s = np.std(sample, ddof=1)
              t_stat = (sample_mean - mu0) / (s / np.sqrt(n))
              if alternative == 'two-sided':
                  p_value = 2 * (1 - t.cdf(abs(t_stat), df=n-1))
              elif alternative == 'less':
                  p_value = t.cdf(t_stat, df=n-1)
              elif alternative == 'greater':
                  p_value = 1 - t.cdf(t_stat, df=n-1)
              else:
                  raise ValueError("alternative must be 'two-sided', 'less', or 'greater'")
              return p_value

      sample1 = np.array([0.53] * 121)

      p_value = mean_hypothesis_test(sample1, mu0=0.50, alternative='greater', std=0.11)

      print(f"P-значение: {p_value:.5f}")

      if p_value < 0.01: #где 0.01 уровень значимости
          print("Отвергаем гипотезу")
      else:
          print("Принимаем гипотезу.")

      P-значение: 0.00135
      Отвергаем гипотезу
```

Задача 2

Проектный контролируемый размер изделий, изготавливаемых станком-автоматом, $a = a_0 = 35$ мм. Измерения 20 случайно отобранных изделий дали следующие результаты:

контролируемый размер x_i	34,8	34,9	35,0	35,1	35,3
частота (число изделий) n_i	2	3	4	6	5

Требуется при уровне значимости 0,05 проверить нулевую гипотезу $H_0: a = a_0 = 35$ при конкурирующей гипотезе $H_1: a \neq 35$.

Реализация без scipy

```
import numpy as np

data = {
    34.8: 2,
    34.9: 3,
    35.0: 4,
    35.1: 6,
    35.3: 5
}

mu0=35

sample2 = np.concatenate([[x] * n for x, n in data.items()])

print(f"Выборка: {sample2}")
print(f"Размер выборки: n = {len(sample2)}")
print(f"Выборочное среднее: {np.mean(sample2):.4f}")
print(f"Выборочное стандартное отклонение: {np.std(sample2, ddof=1):.4f}")

Выборка: [34.8 34.8 34.9 34.9 34.9 35.  35.  35.  35.  35.1 35.1 35.1 35.1 35.1
 35.1 35.3 35.3 35.3 35.3 35.3]
Размер выборки: n = 20
Выборочное среднее: 35.0700
Выборочное стандартное отклонение: 0.1658

p_value = mean_hypothesis_test(sample2, mu0, alternative='two-sided', std=None)
#конкурирующая гипотеза a != 35

print(f"Р-значение (написанная функция): {p_value:.5f}")

if p_value < 0.05:
    print("Отвергаем гипотезу")
else:
    print("Принимаем гипотезу.")

Р-значение (написанная функция): 0.07430
Принимаем гипотезу.
```

Реализация scipy

```
import numpy as np

data = {
    34.8: 2,
    34.9: 3,
    35.0: 4,
    35.1: 6,
    35.3: 5
}

mu0=35

sample = np.concatenate([x * n for x, n in data.items()])

print(f"Выборка: {sample2}")
print(f"Размер выборки: n = {len(sample2)}")
print(f"Выборочное среднее: {np.mean(sample2):.4f}")
print(f"Выборочное стандартное отклонение: {np.std(sample2, ddof=1):.4f}")

Выборка: [34.8 34.8 34.9 34.9 34.9 35.  35.  35.  35.  35.1 35.1 35.1 35.1 35.1
 35.1 35.3 35.3 35.3 35.3 35.3]
Размер выборки: n = 20
Выборочное среднее: 35.0700
Выборочное стандартное отклонение: 0.1658


from scipy.stats import ttest_1samp

t_stat, p_value_scipy = ttest_1samp(sample2, popmean=mu0)

print(f"Р-значение (scipy.stats): {p_value_scipy:.5f}")

if p_value < 0.05:
    print("Отвергаем гипотезу")
else:
    print("Принимаем гипотезу.")

Р-значение (scipy.stats): 0.07430
Принимаем гипотезу.
```

Код на колаб:

https://colab.research.google.com/drive/1X8881RsWGYwzBSGnF_e3aXPtJWKZxFob?usp=sharing