

Лабораторная работа 6

Тестирование гипотезы о значимости различия средних

Гузовская Александра Чеславовна
Б9123-01.03.02сп

27 апреля 2025

Проверка гипотезы о равенстве двух генеральных средних

При независимых выборках для известных дисперсий

Даны две независимые выборки

X_1, X_2, \dots, X_n из $X \sim N(\mu_X, \sigma_X^2)$ и Y_1, Y_2, \dots, Y_m из $Y \sim N(\mu_Y, \sigma_Y^2)$

с известными среднеквадратическими отклонениями σ_X и σ_Y

Проверяем нулевую гипотезу $H_0 : \mu_X = \mu_Y$ против одной из альтернатив:

- $H_1 : \mu_X > \mu_Y$ (правосторонняя)
- $H_1 : \mu_X < \mu_Y$ (левосторонняя)
- $H_1 : \mu_X \neq \mu_Y$ (двусторонняя)

Статистика

$$Z = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{\sigma_X^2}{n} + \frac{\sigma_Y^2}{m}}}$$

Правило принятия решения:

$$p = \begin{cases} F_Z(z), & \text{для } H_1 : \mu_X < \mu_Y \\ 1 - F_Z(z), & \text{для } H_1 : \mu_X > \mu_Y \\ 2 F_Z(-|z|), & \text{для } H_1 : \mu_X \neq \mu_Y \end{cases}$$

При независимых выборках, когда дисперсии неизвестны

Даны две независимые выборки

X_1, X_2, \dots, X_n из $X \sim N(\mu_X, \sigma_X^2)$ и Y_1, Y_2, \dots, Y_m из $Y \sim N(\mu_Y, \sigma_Y^2)$

Проверяем нулевую гипотезу $H_0 : \mu_X = \mu_Y$ против одной из альтернатив:

- $H_1 : \mu_X > \mu_Y$ (правосторонняя)
- $H_1 : \mu_X < \mu_Y$ (левосторонняя)
- $H_1 : \mu_X \neq \mu_Y$ (двусторонняя)

Статистика

$$T_{n+m-2} = \frac{\bar{X} - \bar{Y}}{\sqrt{\hat{\sigma}_X^2 \cdot (n-1) + \hat{\sigma}_Y^2 \cdot (m-1)}} \cdot \sqrt{\frac{nm(n+m-2)}{n+m}} \approx T(n+m-2)$$

р-значение:

$$p = \begin{cases} F_{T_{n+m-2}}(t), & \text{для } H_1 : \mu_X < \mu_Y \\ 1 - F_{T_{n+m-2}}(t), & \text{для } H_1 : \mu_X > \mu_Y \\ 2 F_{T_{n+m-2}}(-|t|), & \text{для } H_1 : \mu_X \neq \mu_Y \end{cases}$$

Для зависимых выборок

Даны две зависимые выборки

X_1, X_2, \dots, X_n из $X \sim N(\mu_X, \sigma_X^2)$ и Y_1, Y_2, \dots, Y_m из $Y \sim N(\mu_Y, \sigma_Y^2)$

Проверяем нулевую гипотезу $H_0 : \mu_X = \mu_Y$ против $H_1 : \mu_X \neq \mu_Y$

Критерий

$$T_{n-1} = \frac{\bar{D} \cdot \sqrt{n}}{\hat{\sigma}_D} \approx T(n-1), \quad \text{где } D = X - Y$$

р-значение $p = F_{T_{n-1}}(-|t|)$

По выборке объема $n=30$ найден средний вес $\bar{x}=130$ г изделий, изготовленных на первом станке; по выборке объема $m=40$ найден средний вес $\bar{y}=125$ г изделий, изготовленных на втором станке. Генеральные дисперсии известны: $D(X)=60$ г², $D(Y)=80$ г². Требуется, при уровне значимости 0,05, проверить нулевую гипотезу $H_0: M(X)=M(Y)$ при конкурирующей гипотезе $M(X) \neq M(Y)$. Предполагается, что случайные величины X и Y распределены нормально и выборки независимы.

Рис. 1: Задача 1

Условия задач

Из двух партий изделий, изготовленных на двух одинаково настроенных станках, извлечены малые выборки, объемы которых $n = 10$ и $m = 12$. Получены следующие результаты:

контролируемый размер					
изделий первого станка	x_i	3,4	3,5	3,7	3,9
частота (число изделий)	n_i	2	3	4	1
контролируемый размер					
изделий второго станка	y_i	3,2	3,4	3,6	
частота	m_i	2	2	8	

Требуется при уровне значимости 0,02 проверить гипотезу $H_0: M(X) = M(Y)$ о равенстве средних размеров изделий при конкурирующей гипотезе $H_1: M(X) \neq M(Y)$. Предполагается, что случайные величины X и Y распределены нормально.

Рис. 2: Задача 2

Двумя приборами в одном и том же порядке измерены шесть деталей и получены следующие результаты измерений (в сотых долях миллиметра):

$$x_1 = 2, x_2 = 3, x_3 = 5, x_4 = 6, x_5 = 8, x_6 = 10;$$

$$y_1 = 10, y_2 = 3, y_3 = 6, y_4 = 1, y_5 = 7, y_6 = 4.$$

При уровне значимости 0,05 установить, значимо или незначимо различаются результаты измерений, в предположении, что они распределены нормально.

Рис. 3: Задача 3

Код программы

```
import numpy as np
from scipy import stats

def compare_means(sample1=None, sample2=None, var1=None, var2=None, dependent=False):
    if sample1 is None or sample2 is None:
        raise ValueError("Нужны обе выборки")

    sample1 = np.asarray(sample1)
    sample2 = np.asarray(sample2)
    n1 = len(sample1)
    n2 = len(sample2)

    if dependent:
        if n1 != n2:
            raise ValueError("Для зависимых выборок размеры должны совпадать")
        diffs = sample1 - sample2
        mean_diff = np.mean(diffs)
        std_diff = np.std(diffs, ddof=1)
        t_stat = mean_diff / (std_diff / np.sqrt(n1))
        df = n1 - 1
        return pvalue_(t_stat, df, alternative, 't')

    mean1 = np.mean(sample1)
    mean2 = np.mean(sample2)

    if var1 is not None and var2 is not None:
        z_stat = (mean1 - mean2) / np.sqrt(var1/n1 + var2/n2)
        return pvalue_(z_stat, None, alternative, 'z')

    std1 = np.std(sample1, ddof=1)
    std2 = np.std(sample2, ddof=1)

    pooled_var = ((n1-1)*std1**2 + (n2-1)*std2**2) / (n1 + n2 - 2)
    t_stat = (mean1 - mean2) / np.sqrt(pooled_var * (1/n1 + 1/n2))
    df = n1 + n2 - 2
```

```

        return pvalue_(t_stat, df, alternative, 't')

def pvalue_(stat, df, alternative, dist_type):
    if dist_type == 't':
        if alternative == 'two-sided':
            return 2 * (1 - stats.t.cdf(abs(stat), df))
        elif alternative == 'less':
            return stats.t.cdf(stat, df)
        else:
            return 1 - stats.t.cdf(stat, df)
    elif alternative == 'two-sided':
        return 2 * (1 - stats.norm.cdf(abs(stat)))
    elif alternative == 'less':
        return stats.norm.cdf(stat)
    else:
        return 1 - stats.norm.cdf(stat)

if __name__ == "__main__":
    print()

    # задача 1
    alpha = 0.05
    sample1 = stats.norm.rvs(loc=130, scale=np.sqrt(60), size=30)
    sample2 = stats.norm.rvs(loc=125, scale=np.sqrt(80), size=40)
    p = compare_means(sample1, sample2, var1=60, var2=80)
    scipy_st, scipy_p = stats.ttest_ind(sample1, sample2, equal_var=False, alternative='less')
    print("Задача 1:", p, "scipy_p = ", scipy_p)
    print(f"{'Отвергаем' if p < alpha else 'Принимаем'} H0 при alpha={alpha}")

    # задача 2
    alpha = 0.02
    values1 = [3.4, 3.5, 3.7, 3.9]
    counts1 = [2, 3, 4, 1]
    sample1 = np.repeat(values1, counts1)
    values2 = [3.2, 3.4, 3.6]
    counts2 = [2, 2, 8]
    sample2 = np.repeat(values2, counts2)
    p = compare_means(sample1, sample2)

```

```

scipy_st, scipy_p = stats.ttest_ind(sample1, sample2, equal_var=False, alternative='two-sided')
print("Задача 2: p = ", p, "scipy_p = ", scipy_p)
print(f"{'Отвергаем' if p < alpha else 'Принимаем'} H0 при alpha={alpha}")

# задача 3
alpha = 0.05
sample1=[2, 3, 5, 6, 8, 10]
sample2=[10, 3, 6, 1, 7, 4]
p = compare_means(sample1, sample2, dependent=True)
scipy_st, scipy_p = stats.ttest_ind(sample1, sample2, alternative='two-sided')
print("Задача 3:", p, "scipy_p = ", scipy_p)
print(f"{'Отвергаем' if p < alpha else 'Принимаем'} H0 при alpha={alpha}")

print()

# ttest_ind(a, b, *, axis=0, equal_var=True, nan_policy='propagate', permutation=1000000)

```

Вывод программы

```

Задача 1: 0.17616480463887063 scipy_p = 0.17400031082647058
Принимаем H0 при alpha=0.05
Задача 2: p = 0.16299851502182805 scipy_p = 0.16459547698225654
Принимаем H0 при alpha=0.02
Задача 3: 0.8165892159697639 scipy_p = 0.7857311649473716
Принимаем H0 при alpha=0.05

```