МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Лабораторная работа №6
Тестирование гипотезы о среднем

Дисциплина «Теория вероятностей и математическая статистика»

Студент группы Б9123-01.03.02ии
Моттуева Уруйдана Михайловна

г. Владивосток

2025

1. Напишите функцию, проверяющую гипотезу о незначимости различий между средними двух нормальных генеральных совокупностей. Функция по переданным параметрам должна учитывать, что: 1) дисперсии могут быть известны, а могут быть неизвестны (в этом случае внимательно прочитайте условия применения критерия), 2) выборки могут быть зависимыми, а могут быть независимыми, 3) критерий может быть право-, лево-, дву-сторонним. Результат функции — p-значение или ошибка в случае возникновения причин невозможности проведения теста. Результаты собственной функции нужно сверить с результатами встроенного метода.

Дано:

2 независимые выборки $X_1, X_2, \ldots, X_n$ из $X \sim N(\mu_X, \sigma_X^2)$ и $Y_1, Y_2, \ldots, Y_m$ из $Y \sim N(\mu_Y, \sigma_Y^2)$, а σ_X и σ_Y известны

Гипотезы:

$H_0\colon \mu_X = \mu_Y$;
$H_1\colon \mu_X \vee \mu_Y$.

Z-тест

Используется, когда:

- Дисперсии генеральных совокупностей известны.
- Выборки независимы.

Z-статистика:

$$Z = \frac{\bar{X} - \bar{Y}}{\sqrt{\left(\frac{\sigma_X^2}{n} + \frac{\sigma_Y^2}{m}\right)}} \sim N(0;1)$$

где:

- $\bar{X}, \bar{Y}$ — выборочные средние,
- $\sigma_X^2,\ \sigma_Y^2$ — известные дисперсии генеральных совокупностей,
- n, m — размеры выборок.

p-значение

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

```python
# Случай 1: известны генеральные дисперсии
if known_variance:
    se = np.sqrt(x_var/n + y_var/m)  # стандартная ошибка
    z = (x_bar - y_bar) / se

    if alternative == 'two-sided':
        p_value = 2 * (1 - stats.norm.cdf(abs(z)))
    elif alternative == 'greater':
        p_value = 1 - stats.norm.cdf(z)
    elif alternative == 'less':
        p_value = stats.norm.cdf(z)ь
    return p_value
```

```python
# функция scipy
se = np.sqrt(x_var/n + y_var/m)
z = (x_bar - y_bar) / se
p_value_scipy = 2 * (1 - stats.norm.cdf(abs(z)))

print(f"Результат scipy: p-value = {p_value_scipy}")
```

F-тест

Дано:

2 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 $Y \sim N(\mu_Y, \sigma_Y^2)$, а σ_X и σ_Y неизвестны

Гипотезы:

$H_0: \sigma_X = \sigma_Y$;
$H_1: \sigma_X \vee \sigma_Y, \ \vee \in \{<, >, \neq\}$.

F-тест Фишера

Если p-значение F-теста больше уровня значимости (α), то дисперсии считаются равными.

Если p-значение F-теста меньше α, то дисперсии считаются неравными.

Статистика:

$$F_{F_{n-1,m-1}} = \frac{\hat{\sigma}_X^2}{\hat{\sigma}_Y^2} \sim F(n-1, m-1).$$

p-значение:

$$p = \begin{cases} F_{F_{n-1,m-1}}(f), & H_1: \sigma_X < \sigma_Y \\ 1 - F_{F_{n-1,m-1}}(f), & H_1: \sigma_X > \sigma_Y \\ 2min\left(F_{F_{n-1,m-1}}(f), 1 - F_{F_{n-1,m-1}}(f)\right), & H_1: \sigma_X \neq \sigma_Y \end{cases}$$

где $F_{F_{n-1,m-1}}(f)$ – функция распределения Фишера с $n-1$ и $m-1$ степенями свободы.

Моя реализация

```
# Проверка равенства дисперсий с помощью F-теста
var_x = np.var(x_sample, ddof=1)
var_y = np.var(y_sample, ddof=1)

if var_x > var_y:
    F = var_x / var_y
    dfn = n - 1 #степени свободы числителя
    dfd = m - 1 #степени свободы знаменателя
else:
    F = var_y / var_x
    dfn = m - 1
    dfd = n - 1

if alternative == 'two-sided':
    p_var = 2 * min(stats.f.cdf(F, dfn, dfd), stats.f.sf(F, dfn, dfd))
    conclusion = f'Дисперсии {"не" if p_var < alpha else ""} равны (p={p_var:.4f})'
elif alternative == 'greater':
    p_var = 1-stats.f.sf(F, dfn, dfd)
    conclusion = f'Дисперсия первой выборки {"значимо >" if p_var < alpha else "не >"} второй (p={p_var:.4f})'
elif alternative == 'less':
    p_var = stats.f.cdf(F, dfn, dfd)
    conclusion = f'Дисперсия первой выборки {"значимо <" if p_var < alpha else "не <"} второй (p={p_var:.4f})'

# Если p-value > 0.05, считаем дисперсии равными
equal_var = (p_var > 0.05)
```

С использованием scipy

```
f_stat = x_var / y_var if x_var > y_var else y_var / x_var
p_value_f = stats.f.cdf(f_stat, n - 1, m - 1) * 2

if p_value_f < 0.05:
    equal_var = False
else:
    equal_var = True
```

Т-тест

Дано:

2 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 $Y \sim N(\mu_Y, \sigma_Y^2)$.

Гипотезы:

$H_0: \mu_X = \mu_Y$;

$H_1: \mu_X \vee \mu_Y$.

Используется, когда:

- Дисперсии неизвестны.
- Выборки независимы.

Т-статистика:

Статистика

$$T_{n+m-2} = \frac{\overline{X} - \overline{Y}}{\sqrt{(n-1)\widehat{\sigma}_X^2 + (m-1)\widehat{\sigma}_Y^2}} \cdot \sqrt{\frac{nm(n+m-2)}{n+m}} \sim T(n+m-2).$$

***p*-значение**

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

где $F_{T_{n+m-2}}$ – функция распределения Стьюдента с $n + m - 2$ степенью свободы.

Моя реализация:

```
# 2) Выборки не зависимы
if paired==False:
    # t-test
    pooled_var = ((n - 1) * var_x + (m - 1) * var_y) / (n + m - 2)
    se = math.sqrt(pooled_var * (1 / n + 1 / m))
    df = n + m - 2  # Степени свободы

    # t-статистика
    t_stat = (x_bar - y_bar) / se

    if alternative == 'greater':
        p_value = stats.t.sf(t_stat, n + m - 2)
    elif alternative == 'less':
        p_value = stats.t.cdf(t_stat, n + m - 2)
    elif alternative == 'two-sided':
        p_value = 2 * stats.t.sf(abs(t_stat), df)
    return p_value
```

С использованием scipy:

```
t_stat, p_value_scipy = stats.ttest_ind(x_sample, y_sample, equal_var=equal_var)
print(f"Результат scipy: p-value = {p_value_scipy}")
```

При зависимых выборках с неизвестными дисперсиями используется критерий:

$$T_{n-1} = \frac{\overline{D}\sqrt{n}}{\hat{\sigma}_D} \sim T(n-1), \text{где } D = X - Y$$

p-значение:

$$p = F_{T_{n-1}}(-|t|)$$

Моя реализация:

```
# 1) Выборки зависимы
if paired:
    d = np.array(x_sample) - np.array(y_sample)
    d_bar = np.mean(d)
    s_d = np.std(d, ddof=1)
    t_stat = d_bar / (s_d / np.sqrt(n))
    df = n - 1

    p_value = 2 * (1 - stats.t.cdf(abs(t_stat), df))

    return p_value
```

С использованием scipy:

```
# функция scipy
t_stat, p_value_scipy = stats.ttest_rel(x_sample, y_sample)
print(f"Результат scipy: p-value = {p_value_scipy}")
```

Ссылка на колаб - https://colab.research.google.com/drive/1WWI8e7dYzIrh9tBrnMaOCenz3fTUZ3ly?usp=sharing

Решение задач

Задача 1

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.

```
x_bar = 130
y_bar = 125
x_var = 60
y_var = 80
n = 30
m = 40
alpha = 0.05
```

```
# функция scipy
se = np.sqrt(x_var/n + y_var/m)
z = (x_bar - y_bar) / se
p_value_scipy = 2 * (1 - stats.norm.cdf(abs(z)))

print(f"Результат scipy: p-value = {p_value_scipy}")
```

```
Результат scipy: p-value = 0.012419330651552318
```

```
p_value_custom1 = compare_means(x_bar=x_bar, y_bar=y_bar, x_var=x_var, y_var=y_var,
                               n=n, m=m, alternative='two-sided', known_variance=True)

print(f"Результат нашей функции: p-value = {p_value_custom1}")

if p_value_custom1 < alpha:
    print(f"Так как p-value ({p_value_custom1}) < {alpha}, отвергаем нулевую гипотезу")
else:
    print(f"Так как p-value ({p_value_custom1}) >= {alpha}, нет оснований отвергать нулевую гипотезу")
```

```
Результат нашей функции: p-value = 0.012419330651552318
Так как p-value (0.012419330651552318) < 0.05, отвергаем нулевую гипотезу
```

Задача 2

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.

```
x_values = np.array([3.4, 3.5, 3.7, 3.9])
x_freq = np.array([2, 3, 4, 1])
n=10

y_values = np.array([3.2, 3.4, 3.6])
y_freq = np.array([2, 2, 8])
m = 12

alpha = 0.02

x_sample = np.repeat(x_values, x_freq)
y_sample = np.repeat(y_values, y_freq)

x_bar = np.mean(x_sample)
y_bar = np.mean(y_sample)

print(f"Среднее первой выборки: {x_bar:.4f}")
print(f"Среднее второй выборки: {y_bar:.4f}")

x_var = np.var(x_sample, ddof=1)
y_var = np.var(y_sample, ddof=1)

print(f"Выборочная дисперсия первой выборки: {x_var:.4f}")
print(f"Выборочная дисперсия второй выборки: {y_var:.4f}")
```

```
Среднее первой выборки: 3.6000
Среднее второй выборки: 3.5000
Выборочная дисперсия первой выборки: 0.0267
Выборочная дисперсия второй выборки: 0.0255
```

```
# функция scipy
f_stat = x_var / y_var if x_var > y_var else y_var / x_var
p_value_f = stats.f.cdf(f_stat, n - 1, m - 1) * 2

if p_value_f < 0.05:
    equal_var = False
else:
    equal_var = True

t_stat, p_value_scipy = stats.ttest_ind(x_sample, y_sample, equal_var=equal_var)
print(f"Результат scipy: p-value = {p_value_scipy}")
```

```
Результат scipy: p-value = 0.16299851502182822
```

```
# Моя функция
p_value_custom2 = compare_means(x_bar=x_bar, y_bar=y_bar, x_var=x_var, y_var=y_var, x_sample=x_sample, y_sample=y_sample,
                               n=n, m=m, paired=False, alternative='two-sided', known_variance=False)

print(f"Результат нашей функции: p-value = {p_value_custom2}")

if p_value_custom2 < alpha:
    print(f"Так как p-value ({p_value_custom2}) < {alpha}, отвергаем нулевую гипотезу")
else:
    print(f"Так как p-value ({p_value_custom2}) >= {alpha}, нет оснований отвергать нулевую гипотезу")
```

```
Результат нашей функции: p-value = 0.16299851502182822
Так как p-value (0.16299851502182822) >= 0.02, нет оснований отвергать нулевую гипотезу
```

Задача 3

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.

```
x_sample = np.array([2, 3, 5, 6, 8, 10])
n=6

y_sample = np.array([10, 3, 6, 1, 7, 4])
m = 6

alpha = 0.02

x_bar = np.mean(x_sample)
y_bar = np.mean(y_sample)

print(f"Среднее первой выборки: {x_bar:.4f}")
print(f"Среднее второй выборки: {y_bar:.4f}")

x_var = np.var(x_sample, ddof=1)
y_var = np.var(y_sample, ddof=1)

print(f"Выборочная дисперсия первой выборки: {x_var:.4f}")
print(f"Выборочная дисперсия второй выборки: {y_var:.4f}")
```

```
Среднее первой выборки: 5.6667
Среднее второй выборки: 5.1667
Выборочная дисперсия первой выборки: 9.0667
Выборочная дисперсия второй выборки: 10.1667
```

```
# функция scipy
t_stat, p_value_scipy = stats.ttest_rel(x_sample, y_sample)
print(f"Результат scipy: p-value = {p_value_scipy}")
```

```
Результат scipy: p-value = 0.8165892159697636
```

```
p_value_custom3 = compare_means(x_bar=x_bar, y_bar=y_bar, x_var=x_var, y_var=y_var, x_sample=x_sample, y_sample=y_sample,
                               n=n, m=m, paired=True, alternative='two-sided', known_variance=False)

# Проверка с помощью scipy
print(f"Результат нашей функции: p-value = {p_value_custom3}")

if p_value_custom3 < alpha:
    print(f"Так как p-value ({p_value_custom3}) < {alpha}, отвергаем нулевую гипотезу")
else:
    print(f"Так как p-value ({p_value_custom3}) >= {alpha}, нет оснований отвергать нулевую гипотезу")
```

```
Результат нашей функции: p-value = 0.8165892159697639
Так как p-value (0.8165892159697639) >= 0.02, нет оснований отвергать нулевую гипотезу
```