摆烂啦.

T_1 $\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$

$$\frac{\bar{X}-\mu}{S/\sqrt{n}} \sim t(n-1)$$

① 假设检验. (α)

$H_0: \mu \le \mu_0$ $H_1: \mu > \mu_0$.

不拒绝 H_0 : $P(\bar{X}-\mu_0 \ge c) \le \alpha$.

$$\Rightarrow P\left(\frac{\bar{X}-\mu_0}{S/\sqrt{n}} \ge \frac{c\sqrt{n}}{S}\right) \le \alpha$$

$$\Rightarrow \bar{X} < \mu_0 + \frac{t_\alpha S}{\sqrt{n}}$$ (不拒绝).

② 置信区间:

$$\frac{\mu_0-\bar{X}}{S/\sqrt{n}} > -t_\alpha$$

即 $\bar{X} < \mu_0 + \frac{t_\alpha S}{\sqrt{n}}$.

也即 ① $\Leftrightarrow$ ②

T_2. $X \sim N(\mu, \sigma^2)$ $\mu_0 = 225$

$H_0: \mu \le 225$ $H_1: \mu > 225$

$\bar{x} = 241.5$ $s = 98.728$ $t = 0.668$

$$P(\bar{X} > \bar{x} \mid H_0) = P\left(\frac{\bar{X} - \mu_0}{s/\sqrt{n}} > \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \,\middle|\, H_0\right)$$

$$= P(T_{n-1} > 0.668)$$

$$\approx 0.257 > \alpha.$$

即无理由.

T_3

$$\frac{\bar{x} - \bar{y}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0, 1)$$

$$\Rightarrow P(\cdot \ \cdot \cdot \cdot) = P(|Z| > 3.7719) = 0.000163 < \alpha.$$

拒绝 H_0.

T_4 全部错误.

<1> <3> p 小不代表不可能.

<2> <4> P是 H_0 下的条件概率 ≠.

<5> 单次决策无概率可言!

<6> P值 ≠ 函数.

T_5 $\chi_0^2 = 0.4617$

$\Rightarrow P(\chi^2(3) \geq \chi_0^2) = 0.925$

即不能拒绝 H_0. 于是 Mendel 是对的.

T_6. <1> $\chi_0^2 = 1$

$P(\chi^2(5) \geq \chi_0^2) = 0.96$

✓

<2> $\chi_0^2 = 10$ $P(\chi^2(5) \geq \chi_0^2) = 0.075$ ✗

<3> 样本量极大时, 小差异多次积累导致统计显著.

T7 $\chi_0^2 = 16.37 > 6.25 = \chi^2(3)_\alpha$ 有.

T8 $\chi_0^2 = 0.5103 < 7.81 = \chi^2(3)_\alpha$ 不认有

T9

```python
import numpy as np
import math

mu = 5
sigma = 1
n = 100
mu0 = 5.2
alpha = 0.05
num_trials = 1000
reject_count_mean = 0
reject_count_edge = 0
p_values_mean = []
p_values_edge = []

for _ in range(num_trials):
    sample = np.random.normal(mu, sigma, n)
    sample_mean = np.mean(sample)
    sample_std = np.std(sample, ddof=1)
    t_stat = (sample_mean - mu0) / (sample_std / np.sqrt(n))
    p_val_mean = 1 - (0.5 * (1 + math.erf(t_stat / math.sqrt(2))))
    p_values_mean.append(p_val_mean)
    if p_val_mean < alpha:
        reject_count_mean += 1

    edge_stat = 0.5 * (sample[0] + sample[-1])
    se_edge = sigma / math.sqrt(2)
    z_edge = (edge_stat - mu0) / se_edge
    p_val_edge = 1 - (0.5 * (1 + math.erf(z_edge / math.sqrt(2))))
    p_values_edge.append(p_val_edge)
    if p_val_edge < alpha:
        reject_count_edge += 1

type1_error_mean = reject_count_mean / num_trials
type1_error_edge = reject_count_edge / num_trials
```

```
Number of rejections (sample mean): 1, Type I error rate: 0.0010
Number of rejections (1/2(X1+Xn)): 29, Type I error rate: 0.0290
Average p-value (sample mean): 0.9143
Average p-value (1/2(X1+Xn)): 0.5780
Press any key to continue . . .
```