🔧 Шаг 1. Сгенерируем выборку

Пусть у нас есть выборка из $n = 200$ наблюдений, сгенерированная из истинного нормального распределения:

$$X_i \sim \mathrm{N}(5,\, 2^2) \quad (i = 1, \ldots, 200).$$

В Python это можно сделать так:

```python
import numpy as np
np.random.seed(42)
sample = np.random.normal(loc=5, scale=2, size=200)
```

🔧 Шаг 2. Оценим параметры (если неизвестны)

Если мы **не знаем** μ и σ , оценим их по выборке:

- $\hat{\mu} = \bar{x} = \frac{1}{n} \sum x_i$
- $\hat{\sigma} = s = \sqrt{\frac{1}{n-1} \sum (x_i - \bar{x})^2}$

В нашем случае:

```python
mu_hat = sample.mean()      # ≈ 5.0
sigma_hat = sample.std(ddof=1)  # ≈ 2.0
```

Теперь **нулевая гипотеза**:

> «Выборка взята из $\mathrm{N}(\hat{\mu}, \hat{\sigma}^2)$ ».

🔧 Шаг 3. Разобьём диапазон на интервалы (k категорий)

- Выберем, например, k = 8 интервалов.
- Убедимся, что **ожидаемая частота в каждом ≥ 5** (условие применимости χ^2).
- Лучше строить интервалы **по квантилям** нормального распределения, чтобы ожидаемые частоты были примерно равны.

Пример: разобьём на 8 интервалов с равными вероятностями (по 12.5% каждый):

```python
from scipy.stats import norm

```

```python
k = 8
probs = np.linspace(0, 1, k + 1)  # [0, 0.125, 0.25, ..., 1]
edges = norm.ppf(probs, loc=mu_hat, scale=sigma_hat)  # границы по квантилям
```

🔧 Шаг 4. Посчитаем наблюдаемые и ожидаемые частоты

- Наблюдаемые O_i : сколько значений попало в каждый интервал.
- Ожидаемые $E_i = n/k = 200/8 = 25$ (так как интервалы равновероятны).

```python
observed, _ = np.histogram(sample, bins=edges)
expected = np.full(k, 200 / k)
```

> Если какие-то $E_i < 5$, объединяем соседние интервалы.

🔧 Шаг 5. Вычислим статистику хи-квадрат

$$\chi^2_{\text{набл}} = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

```python
chi2_stat = ((observed - expected) ** 2 / expected).sum()
```

Допустим, получили: $\chi^2_{\text{набл}} = 5.1$.

🔧 Шаг 6. Определим число степеней свободы

Общая формула:

$$\nu = k - 1 - r,$$

где:

- k — число интервалов,
- r — число параметров, **оценённых по выборке**.

У нас: $k = 8$, $r = 2$ (оценили μ и σ), значит

$$\nu = 8 - 1 - 2 = 5.$$

🔧 Шаг 7. Примем решение

- Выберем уровень значимости $\alpha = 0.05$.
- Критическое значение: $\chi^2_{0.95}(5) \approx 11.07$ (из таблицы или scipy).

```python
from scipy.stats import chi2
p_value = 1 - chi2.cdf(chi2_stat, df=5)
```

Если:

- $\chi^2_{\text{набл}} = 5.1 < 11.07 \rightarrow$ не отвергаем H_0,
- $p\text{-value} \approx 0.40 > 0.05 \rightarrow$ согласие с нормальным распределением **не противоречит данным**.

📊 Шаг 8. Визуализация (гистограмма + теоретическая плотность)

```python
import matplotlib.pyplot as plt

x = np.linspace(sample.min(), sample.max(), 200)
plt.hist(sample, bins=20, density=True, alpha=0.6, label='Выборка')
plt.plot(x, norm.pdf(x, mu_hat, sigma_hat), 'r-', lw=2, label='Теоретическая
plt.legend()
plt.title('Гистограмма и нормальная плотность')
plt.show()
```

На графике будет видно, что гистограмма **хорошо ложится** на кривую нормального распределения — визуальное подтверждение вывода критерия.

✅ Вывод:

Для выборки, сгенерированной из нормального распределения, **критерий Пирсона не отвергает гипотезу** о нормальности (что логично).
Если бы выборка была, скажем, из экспоненциального распределения, критерий **отверг бы** H_0.