Module 6: Hypothesis testing for health data

Data analysis with Python for health specialists

Yaé Ulrich Gaba

2026

AIRINA Labs

The logic of hypothesis testing

Every clinical question $\rightarrow$ two competing statements:

- H_0 (null): There is no effect / no difference / no association.
- H_1 (alternative): There is an effect / a difference / an association.

The 5 steps:

1. State hypotheses (H_0 and H_1)
2. Choose significance level α (usually 0.05)
3. Collect data, compute test statistic
4. Compute p-value
5. Decide: if $p < \alpha$, reject H_0

Clinical framing

A statistician writes $H_0 : \mu_1 = \mu_2$. A clinician writes "no difference in SBP between statin and placebo groups." Both say the same thing.

p-values: what they mean and what they don't

A p-value IS: the probability of seeing data this extreme if H_0 is true.

A p-value is NOT:

- The probability that H_0 is true
- The probability the result is "due to chance"
- A measure of effect size
- Meaningfully different at 0.049 vs. 0.051

The big mistake
A study with 100 000 patients can find $p < 0.001$ for a BP difference of 0.5 mmHg — statistically significant but clinically meaningless. Always report **effect size** alongside p-values.

One-sample t-test

Q: Is mean cholesterol in our clinic different from the national average of 200 mg/dL?

```
from scipy import stats
import numpy as np

np.random.seed(42)
cholesterol = np.random.normal(loc=212, scale=35, size=80)

t_stat, p_value = stats.ttest_1samp(cholesterol, popmean=200)
print(f"Sample mean: {cholesterol.mean():.1f} mg/dL")
print(f"t = {t_stat:.3f}, p = {p_value:.4f}")
```

Compares a **sample mean** to a **known reference value** (guideline threshold, national average).

Two-sample t-test

Q: Is systolic BP different between treatment and control groups?

```
np.random.seed(42)
treatment = np.random.normal(loc=128, scale=15, size=60)
control = np.random.normal(loc=138, scale=16, size=55)

# Welch's t-test (safer default: no equal variance assumption)
t_stat, p_value = stats.ttest_ind(treatment, control,
                                  equal_var=False)
print(f"Difference: {control.mean()-treatment.mean():.1f} mmHg")
print(f"p = {p_value:.4f}")
```

Check assumptions

Normality: `stats.shapiro()`. Equal variances: `stats.levene()`. If variances differ, use Welch's t-test (`equal_var=False`).

Paired t-test

Q: Does a 12-week exercise program reduce blood pressure?

```
np.random.seed(42)
bp_before = np.random.normal(loc=142, scale=12, size=45)
bp_after = bp_before - np.random.normal(loc=8, scale=6, size=45)

t_stat, p_value = stats.ttest_rel(bp_before, bp_after)
diffs = bp_before - bp_after
print(f"Mean reduction: {diffs.mean():.1f} mmHg")
print(f"p = {p_value:.6f}")
```

Common mistake
Using an **independent** t-test on before/after data ignores the pairing and loses statistical power. Same patients measured twice → always use a **paired** t-test.

Chi-square test of independence

Q: Is diabetes prevalence associated with obesity category?

```
# Build contingency table
contingency = pd.crosstab(df["bmi_category"],
                          df["diabetes_label"])

# Chi-square test
chi2, p_value, dof, expected = stats.chi2_contingency(
    contingency)
print(f"Chi2 = {chi2:.2f}, dof = {dof}, p = {p_value:.4f}")
```

Works with **categorical** variables. Requires expected cell counts ≥ 5. For small cells, use Fisher's exact test: `stats.fisher_exact(table)`.

Mann-Whitney U test

When data is not normally distributed (length of stay, costs, small samples):

```
np.random.seed(42)
los_surgical = np.random.exponential(scale=7, size=40)
los_medical = np.random.exponential(scale=4.5, size=45)

u_stat, p_value = stats.mannwhitneyu(
    los_surgical, los_medical, alternative="two-sided")
print(f"Median surgical: {np.median(los_surgical):.1f} days")
print(f"Median medical:  {np.median(los_medical):.1f} days")
print(f"p = {p_value:.4f}")
```

Rule of thumb: Normal → t-test. Non-normal → Mann-Whitney. Categorical → chi-square. Paired non-normal → Wilcoxon signed-rank.

Multiple testing correction

Testing 20 hypotheses at $\alpha = 0.05 \rightarrow$ expect 1 false positive even if nothing is going on.

```
from statsmodels.stats.multitest import multipletests

# Bonferroni: divide alpha by number of tests
rejected_bonf, _, _, _ = multipletests(
    p_values, alpha=0.05, method="bonferroni")

# FDR (Benjamini-Hochberg): less conservative
rejected_fdr, _, _, _ = multipletests(
    p_values, alpha=0.05, method="fdr_bh")
```

Bonferroni: strict, minimizes false positives (confirmatory trials).

FDR: less strict, controls false discovery rate (exploratory/genomics).

Effect size: Cohen's d

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{\text{pooled}}}$$

- $|d| < 0.2$: negligible
- $0.2 \leq |d| < 0.5$: small
- $0.5 \leq |d| < 0.8$: medium
- $|d| \geq 0.8$: large

```
def cohens_d(g1, g2):
    n1, n2 = len(g1), len(g2)
    pooled = np.sqrt(((n1-1)*g1.var(ddof=1) +
                      (n2-1)*g2.var(ddof=1)) / (n1+n2-2))
    return (g1.mean() - g2.mean()) / pooled
```

Always report effect size alongside p-values. Statistical significance $\neq$ clinical significance.

What you can do after this module

1. Formulate H_0 / H_1 for clinical research questions
2. Apply the right test: t-test (1-sample, 2-sample, paired), chi-square, Mann-Whitney
3. Interpret p-values correctly and avoid common misconceptions
4. Correct for multiple comparisons (Bonferroni, FDR)
5. Compute Cohen's *d* to assess clinical meaningfulness

Next: Module 7 — Regression for health outcomes

Exercises (Hour 3)

1. **One-sample t-test:** Framingham cholesterol vs. 200 mg/dL
2. **Two-sample t-test:** Systolic BP in CHD vs. non-CHD groups with Cohen's *d*
3. **Chi-square:** Smoking vs. 10-year CHD risk
4. **Mini-project:** Run all tests on the Framingham dataset with multiple testing correction; identify the variable with the largest effect size