1. What is hypothesis testing in statistics?

Answer:
Hypothesis testing is a statistical method used to make decisions or inferences about population parameters based on sample data. It involves testing an assumption (hypothesis) to determine if there is enough evidence to reject it.

2. What is the null hypothesis, and how does it differ from the alternative hypothesis?

Answer:

- **Null Hypothesis (H_0):** Assumes no effect or no difference.
- **Alternative Hypothesis (H_1):** Represents the opposite of the null; indicates a significant effect or difference.

3. What is the significance level in hypothesis testing, and why is it important?

Answer:
The **significance level (α)** is the threshold for rejecting the null hypothesis, commonly set at 0.05. It defines the probability of making a Type I error (rejecting a true null hypothesis).

4. What does a P-value represent in hypothesis testing?

Answer:
The **p-value** is the probability of obtaining test results at least as extreme as those observed, assuming the null hypothesis is true.

5. How do you interpret the P-value in hypothesis testing?

Answer:

- **If p-value < α**: Reject H_0 (significant result)
- **If p-value ≥ α**: Fail to reject H_0 (not significant)

6. What are Type 1 and Type 2 errors in hypothesis testing?

Answer:

- **Type I Error (α):** Rejecting a true H_0.
- **Type II Error (β):** Failing to reject a false H_0.

7. What is the difference between a one-tailed and a two-tailed test in hypothesis testing?

Answer:

- **One-tailed:** Tests if the parameter is greater or less than a value.
- **Two-tailed:** Tests if the parameter is **not equal** to a value.

8. What is the Z-test, and when is it used in hypothesis testing?

Answer:
The **Z-test** is used to determine if there is a significant difference between sample and population means when population standard deviation is known and sample size is large ($n \geq 30$).

9. How do you calculate the Z-score, and what does it represent in hypothesis testing?

Answer:

Z
=
x
ˉ
−
μ
σ
/
n
$Z = \sigma / n \; \bar{x} - \mu$
Where

x
ˉ

$\bar{x}$ = sample mean,

μ
μ = population mean,

σ
σ = population SD, n = sample size. It tells how many standard deviations the sample mean is from the population mean.

10. What is the T-distribution, and when should it be used instead of the normal distribution?

Answer:
The **t-distribution** is used when the sample size is small ($n < 30$) and population standard deviation is unknown. It is wider and has heavier tails than the normal distribution.

11. What is the difference between a Z-test and a T-test?

Answer:

- **Z-test:** Known population SD, large sample.
- **T-test:** Unknown population SD, small sample.

12. What is the T-test, and how is it used in hypothesis testing?

Answer:
A **T-test** evaluates whether the means of two groups are statistically different. Types:

- One-sample
- Independent two-sample
- Paired-sample

13. What is the relationship between Z-test and T-test in hypothesis testing?

Answer:
Both test hypotheses about means. T-test converges to Z-test as sample size increases.

14. What is a confidence interval, and how is it used to interpret statistical results?

Answer:
A **confidence interval (CI)** is a range of values likely to contain a population parameter. A 95% CI means we are 95% confident the true value lies within that range.

15. What is the margin of error, and how does it affect the confidence interval?

Answer:
Margin of Error (MoE) quantifies the uncertainty in estimate. Larger MoE = wider CI; affected by sample size and variability.

16. How is Bayes' Theorem used in statistics, and what is its significance?

Answer:
Bayes' Theorem is used to update probabilities based on new evidence. It's central to Bayesian statistics and decision-making under uncertainty.

17. What is the Chi-square distribution, and when is it used?

Answer:
A **Chi-square distribution** is used for categorical data analysis, particularly:

- Goodness-of-fit test
- Test of independence

18. What is the Chi-square goodness of fit test, and how is it applied?

Answer:
It checks if observed frequencies match expected frequencies for a categorical variable.

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

$\chi 2=\sum E(O-E)2$

19. What is the F-distribution, and when is it used in hypothesis testing?

Answer:
The **F-distribution** is used to compare two variances, especially in ANOVA and F-tests. It's positively skewed and defined by two degrees of freedom.

20. What is an ANOVA test, and what are its assumptions?

Answer:
ANOVA (Analysis of Variance) tests if means across multiple groups differ significantly.

Assumptions:

- Normality
- Equal variances (homogeneity)
- Independence

21. What are the different types of ANOVA tests?

Answer:

- **One-way ANOVA:** One independent variable
- **Two-way ANOVA:** Two independent variables
- **Repeated measures ANOVA:** Same subjects across conditions

22. What is the F-test, and how does it relate to hypothesis testing?

Answer:
The **F-test** compares two variances. It's used in ANOVA and regression to test if model variables are significant.

1. Write a Python program to perform a Z-test for comparing a sample mean to a known population mean and interpret the results

python
CopyEdit

```
from scipy.stats import norm
import math

# Sample data
sample_mean = 52
population_mean = 50
population_std = 5
n = 30

# Z-score calculation
z = (sample_mean - population_mean) / (population_std /
math.sqrt(n))
p_value = 2 * (1 - norm.cdf(abs(z)))

print(f"Z-score: {z:.2f}")
print(f"P-value: {p_value:.4f}")

if p_value < 0.05:
    print("Reject the null hypothesis")
else:
    print("Fail to reject the null hypothesis")
```

2. Simulate random data to perform hypothesis testing and calculate the corresponding P-value using Python

python
CopyEdit
```
import numpy as np
from scipy.stats import ttest_1samp

# Simulate data
np.random.seed(0)
data = np.random.normal(loc=50, scale=5, size=30)

# Perform t-test against population mean of 52
t_stat, p_val = ttest_1samp(data, 52)

print(f"T-statistic: {t_stat:.2f}")
print(f"P-value: {p_val:.4f}")
```

3. Implement a one-sample Z-test using Python to compare the sample mean with the population mean

python
CopyEdit
```
from scipy.stats import norm

def one_sample_z_test(sample_mean, population_mean,
population_std, n):
    z = (sample_mean - population_mean) / (population_std /
math.sqrt(n))
    p_value = 2 * (1 - norm.cdf(abs(z)))
    return z, p_value

z, p = one_sample_z_test(53, 50, 4, 40)
print(f"Z-score: {z:.2f}, P-value: {p:.4f}")
```

4. Perform a two-tailed Z-test using Python and visualize the decision region on a plot

python
CopyEdit
```
import matplotlib.pyplot as plt
import numpy as np

# Z-score
z = 2.0
x = np.linspace(-4, 4, 1000)
y = norm.pdf(x)

plt.plot(x, y, label='Normal Distribution')
```

```
plt.fill_between(x, y, where=(x <= -1.96) | (x >= 1.96),
color='red', alpha=0.3, label='Rejection region (α=0.05)')
plt.axvline(z, color='blue', linestyle='--', label=f'Z =
{z}')
plt.title("Two-Tailed Z-Test Decision Region")
plt.legend()
plt.show()
```

5. Create a Python function that calculates and visualizes Type 1 and Type 2 errors during hypothesis testing

python
CopyEdit
```
def visualize_errors(mu0=0, mu1=1, sigma=1, alpha=0.05):
    x = np.linspace(-4, 4, 1000)
    y0 = norm.pdf(x, mu0, sigma)
    y1 = norm.pdf(x, mu1, sigma)

    z_crit = norm.ppf(1 - alpha / 2)

    plt.plot(x, y0, label='H0 distribution')
    plt.plot(x, y1, label='H1 distribution')

    plt.fill_between(x, y0, where=(x > z_crit) | (x <
-z_crit), color='red', alpha=0.3, label='Type I Error')
    plt.fill_between(x, y1, where=(x < z_crit) & (x >
-z_crit), color='blue', alpha=0.3, label='Type II Error')

    plt.legend()
    plt.title("Type I and Type II Errors")
    plt.show()

visualize_errors()
```

6. Write a Python program to perform an independent T-test and interpret the results

python
CopyEdit
```
from scipy.stats import ttest_ind

group1 = [22, 24, 25, 28, 30]
group2 = [30, 32, 33, 31, 29]

t_stat, p_value = ttest_ind(group1, group2)
```

```
print(f"T-statistic: {t_stat:.2f}, P-value: {p_value:.4f}")
```

7. Perform a paired sample T-test using Python and visualize the comparison results

python
CopyEdit

```
from scipy.stats import ttest_rel

before = [85, 88, 90, 87, 86]
after = [87, 90, 91, 89, 88]

t_stat, p_value = ttest_rel(before, after)
print(f"Paired T-test statistic: {t_stat:.2f}, P-value:
{p_value:.4f}")

plt.plot(before, label='Before')
plt.plot(after, label='After')
plt.title("Paired Sample Comparison")
plt.legend()
plt.show()
```

8. Simulate data and perform both Z-test and T-test, then compare the results using Python

python
CopyEdit

```
sample = np.random.normal(50, 10, 30)

# Z-test (assuming known population std)
z = (np.mean(sample) - 50) / (10 / np.sqrt(30))
p_z = 2 * (1 - norm.cdf(abs(z)))

# T-test
t_stat, p_t = ttest_1samp(sample, 50)

print(f"Z-test: z={z:.2f}, p={p_z:.4f}")
print(f"T-test: t={t_stat:.2f}, p={p_t:.4f}")
```

9. Write a Python function to calculate the confidence interval for a sample mean and explain its significance

python
CopyEdit

```
import scipy.stats as stats
```

```
def confidence_interval(data, confidence=0.95):
    mean = np.mean(data)
    sem = stats.sem(data)
    margin = sem * stats.t.ppf((1 + confidence) / 2.,
len(data) - 1)
    return (mean - margin, mean + margin)

sample_data = [52, 55, 54, 53, 51, 56]
ci = confidence_interval(sample_data)
print(f"95% Confidence Interval: {ci}")
```

PRACTICAL QUESTION AND ANSWERS

1. Write a Python program to calculate the margin of error for a given confidence level using sample data

python
CopyEdit

```
import numpy as np
import scipy.stats as stats

data = [50, 52, 48, 47, 53, 51, 49]
confidence = 0.95
n = len(data)
mean = np.mean(data)
sem = stats.sem(data)
margin_error = sem * stats.t.ppf((1 + confidence) / 2., n -
1)

print(f"Margin of Error: {margin_error:.2f}")
```

2. Implement a Bayesian inference method using Bayes' Theorem in Python and explain the process

python
CopyEdit

```
def bayes_theorem(prior, likelihood, evidence):
    posterior = (likelihood * prior) / evidence
    return posterior

# Example: Medical test
P_D = 0.01          # Prior: probability of disease
P_Pos_D = 0.99      # Likelihood: true positive
P_Pos = 0.05        # Evidence: overall positive test rate

posterior = bayes_theorem(P_D, P_Pos_D, P_Pos)
```

```
print(f"Posterior probability of disease given a positive test: {posterior:.4f}")
```

3. Perform a Chi-square test for independence between two categorical variables in Python

python
CopyEdit
```
import pandas as pd
from scipy.stats import chi2_contingency

data = [[20, 15], [30, 35]]
chi2, p, dof, expected = chi2_contingency(data)

print(f"Chi-square Statistic: {chi2:.2f}")
print(f"P-value: {p:.4f}")
```

4. Write a Python program to calculate the expected frequencies for a Chi-square test based on observed data

python
CopyEdit
```
obs = np.array([[20, 15], [30, 35]])
_, _, _, expected = chi2_contingency(obs)
print("Expected frequencies:\n", expected)
```

5. Perform a goodness-of-fit test using Python to compare the observed data to an expected distribution

python
CopyEdit
```
from scipy.stats import chisquare

observed = [25, 30, 45]
expected = [33.3, 33.3, 33.3]

chi2_stat, p_val = chisquare(f_obs=observed, f_exp=expected)
print(f"Chi-square Statistic: {chi2_stat:.2f}, P-value: {p_val:.4f}")
```

6. Create a Python script to simulate and visualize the Chi-square distribution and discuss its characteristics

python
CopyEdit
```
import matplotlib.pyplot as plt
```

```
df = 4
x = np.linspace(0, 20, 500)
y = stats.chi2.pdf(x, df)

plt.plot(x, y, label=f'Chi-square Distribution df={df}')
plt.title("Chi-square Distribution")
plt.xlabel("Value")
plt.ylabel("Probability Density")
plt.legend()
plt.grid()
plt.show()
```

7. Implement an F-test using Python to compare the variances of two random samples

python
CopyEdit
```
group1 = [20, 22, 19, 21, 20]
group2 = [30, 31, 29, 35, 28]

var1 = np.var(group1, ddof=1)
var2 = np.var(group2, ddof=1)

F = var1 / var2
df1, df2 = len(group1) - 1, len(group2) - 1
p_val = 1 - stats.f.cdf(F, df1, df2)

print(f"F-statistic: {F:.2f}, P-value: {p_val:.4f}")
```

8. Write a Python program to perform an ANOVA test to compare means between multiple groups and interpret the results

python
CopyEdit
```
from scipy.stats import f_oneway

a = [25, 30, 28, 32, 27]
b = [35, 40, 38, 37, 36]
c = [45, 48, 46, 50, 47]

f_stat, p_value = f_oneway(a, b, c)
print(f"F-statistic: {f_stat:.2f}, P-value: {p_value:.4f}")
```

9. Perform a one-way ANOVA test using Python to compare the means of different groups and plot the results

python
CopyEdit
```
import seaborn as sns

data = pd.DataFrame({
    'Scores': a + b + c,
    'Group': ['A']*5 + ['B']*5 + ['C']*5
})

sns.boxplot(x='Group', y='Scores', data=data)
plt.title("One-way ANOVA Boxplot")
plt.show()
```

10. Write a Python function to check the assumptions (normality, independence, and equal variance) for ANOVA

python
CopyEdit
```
from scipy.stats import shapiro, levene

def check_anova_assumptions(*groups):
    for i, g in enumerate(groups):
        stat, p = shapiro(g)
        print(f"Group {i+1} Normality p-value: {p:.4f}")
    stat, p = levene(*groups)
    print(f"Equal Variance p-value (Levene’s test): {p:.4f}")

check_anova_assumptions(a, b, c)
```

11. Perform a two-way ANOVA test using Python to study the interaction between two factors and visualize the results

python
CopyEdit
```
import statsmodels.api as sm
from statsmodels.formula.api import ols

df = pd.DataFrame({
    'score': [85, 90, 88, 92, 95, 78, 85, 89, 84, 91],
    'gender': ['M', 'M', 'M', 'F', 'F', 'M', 'F', 'F', 'M',
'F'],
    'method': ['A', 'B', 'A', 'B', 'A', 'A', 'B', 'B', 'A',
'A']
})
```

```
model = ols('score ~ C(gender) + C(method) +
C(gender):C(method)', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
```

12. Write a Python program to visualize the F-distribution and discuss its use in hypothesis testing

python
CopyEdit

```
dof1, dof2 = 5, 10
x = np.linspace(0, 5, 500)
y = stats.f.pdf(x, dof1, dof2)

plt.plot(x, y)
plt.title(f"F-distribution (df1={dof1}, df2={dof2})")
plt.xlabel("F value")
plt.ylabel("Density")
plt.grid()
plt.show()
```

13. Perform a one-way ANOVA test in Python and visualize the results with boxplots to compare group means

(Already covered in Q8 and Q9)

14. Simulate random data from a normal distribution, then perform hypothesis testing to evaluate the means

python
CopyEdit

```
data = np.random.normal(loc=50, scale=5, size=100)
t_stat, p_val = ttest_1samp(data, 52)
print(f"T-statistic: {t_stat:.2f}, P-value: {p_val:.4f}")
```

15. Perform a hypothesis test for population variance using a Chi-square distribution and interpret the results

python
CopyEdit

```
sample = [52, 50, 48, 49, 53]
n = len(sample)
sample_var = np.var(sample, ddof=1)
hyp_var = 4
```

```
chi2_stat = (n - 1) * sample_var / hyp_var
p_val = 1 - stats.chi2.cdf(chi2_stat, n - 1)
print(f"Chi-square stat: {chi2_stat:.2f}, P-value:
{p_val:.4f}")
```

16. Write a Python script to perform a Z-test for comparing proportions between two datasets or groups

python
CopyEdit

```
from statsmodels.stats.proportion import proportions_ztest

successes = [45, 40]
nobs = [100, 100]

stat, pval = proportions_ztest(successes, nobs)
print(f"Z-statistic: {stat:.2f}, P-value: {pval:.4f}")
```

17. Implement an F-test for comparing the variances of two datasets, then interpret and visualize the results

python
CopyEdit

```
group1 = np.random.normal(50, 5, 30)
group2 = np.random.normal(50, 10, 30)

f_stat = np.var(group1, ddof=1) / np.var(group2, ddof=1)
df1, df2 = 29, 29
p_val = 1 - stats.f.cdf(f_stat, df1, df2)

print(f"F-statistic: {f_stat:.2f}, P-value: {p_val:.4f}")
```

18. Perform a Chi-square test for goodness of fit with simulated data and analyze the results

python
CopyEdit

```
observed = [18, 22, 20]
expected = [20, 20, 20]

chi2_stat, p = chisquare(observed, f_exp=expected)
print(f"Chi-square: {chi2_stat:.2f}, P-value: {p:.4f}")
```