Question 1: What is hypothesis testing in statistics?

Ans. Hypothesis testing is a statistical method used to make decisions or inferences about a population parameter based on a sample of data.

It helps us decide whether there is enough evidence to **accept or reject a claim (hypothesis)** about the population.

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

Ans. **Null Hypothesis (H_0):**

- It is the **default assumption** that there is **no effect, no difference, or no relationship** in the population.
- It represents the “status quo” or what we assume to be true until proven otherwise.

🔹 Example:

- A medicine has **no effect** on curing a disease.
- The average exam score of a class = 70.

Question 3: Explain the significance level in hypothesis testing and its role in deciding the outcome of a test.

Ans. The **significance level (α)** is the threshold (a cutoff probability) that we set **before testing** to decide whether to reject the null hypothesis (H_0).

It represents the probability of making a **Type I error** → rejecting H_0 when it is actually true.

Common Values of α

- **0.05 (5%)** → most common
- 0.01 (1%) → stricter
- 0.10 (10%) → more lenient

Role in Hypothesis Testing

1. **Set α before the test**
   - Example: $\alpha = 0.05$ means we accept a 5% chance of wrongly rejecting H_0.
2. **Compare p-value with α**
   - If **p-value < α** → Reject H_0 (evidence supports H_1).
   - If **p-value ≥ α** → Fail to reject H_0 (not enough evidence).
3. **Decision making**
   - α acts as the "cutoff line" that decides if results are **statistically significant**.

Question 4: What are Type I and Type II errors? Give examples of each.

Ans. **Type I and Type II Errors**

When we make decisions in hypothesis testing, errors can occur because we are working with samples, not the whole population.

---

1. Type I Error (False Positive)

- Happens when we **reject the null hypothesis (H_0) even though it is true.**
- It's like a **false alarm.**
- Probability of Type I error = **α (significance level).**

🔹 **Example:**

- A COVID test says a healthy person has the disease.
- Courtroom analogy: An innocent person is declared guilty.
- Research: Claiming a new drug works when it actually doesn't.

---

2. Type II Error (False Negative)

- Happens when we **fail to reject the null hypothesis (H_0) even though it is false.**
- It's like **missing something real.**
- Probability of Type II error = **β.**
- (The power of a test = 1 – β, ability to detect a true effect).

🔹 **Example:**

- A COVID test says a sick person is healthy.
- Courtroom analogy: A guilty person is declared innocent.
- Research: Concluding a new drug doesn't work, when it actually does.

Question 5: What is the difference between a Z-test and a T-test? Explain when to use each

Ans. **Z-test vs T-test**

Both are statistical tests used in **hypothesis testing** to compare sample data with population data (or between groups).
The key difference lies in **sample size** and whether the **population standard deviation (σ)** is known.

---

1. Z-test

- Used when:
  The population standard deviation (σ) is **known**
  The sample size is **large ($n > 30$)** (Central Limit Theorem applies)
- Based on the **Standard Normal Distribution (Z-distribution, mean = 0, SD = 1)**

🔹 **Examples:**

- Checking if the average height of 1000 students = 160 cm, when σ is known.
- Quality control in manufacturing (large samples, known σ).

---

2. T-test

- Used when:
  The population standard deviation (σ) is **unknown**
  The sample size is **small (n ≤ 30)**

- Based on the **Student's t-distribution** (heavier tails than normal distribution, accounts for extra uncertainty in small samples).

🔹 **Examples:**

- Testing if the average marks of 20 students = 70 (σ unknown).

- Comparing the mean weight of two small groups

Question 6: Write a Python program to generate a binomial distribution with n=10 and p=0.5, then plot its histogram.

Ans.import numpy as np

```
import matplotlib.pyplot as plt


# Parameters

n = 10     # number of trials

p = 0.5    # probability of success

size = 1000  # number of samples


# Generate binomial distribution data

data = np.random.binomial(n, p, size)


# Plot histogram

plt.hist(data, bins=range(n+2), edgecolor='black', alpha=0.7)

plt.title("Binomial Distribution (n=10, p=0.5)")

plt.xlabel("Number of Successes")

plt.ylabel("Frequency")

plt.show()
```

Explanation

1. `np.random.binomial(n, p, size)` → generates random samples from a binomial distribution.
    - `n = 10` → 10 trials
    - `p = 0.5` → probability of success in each trial
    - `size = 1000` → number of simulated experiments
2. `plt.hist()` → draws histogram of the generated data.
    - `bins=range(n+2)` ensures bins are aligned with integer success counts.
3. The histogram will be **symmetric around 5**, since with `n=10` and `p=0.5`, the expected number of successes = `n*p = 5`

Question 7: Implement hypothesis testing using Z-statistics for a sample dataset in Python. Show the Python code and interpret the results. sample_data = [49.1, 50.2, 51.0, 48.7, 50.5, 49.8, 50.3, 50.7, 50.2, 49.6, 50.1, 49.9, 50.8, 50.4, 48.9, 50.6, 50.0, 49.7, 50.2, 49.5, 50.1, 50.3, 50.4, 50.5, 50.0, 50.7, 49.3, 49.8, 50.2, 50.9, 50.3, 50.4, 50.0, 49.7, 50.5, 49.9]

Ans. import numpy as np

from scipy.stats import norm

Sample data

sample_data = [49.1, 50.2, 51.0, 48.7, 50.5, 49.8, 50.3, 50.7, 50.2, 49.6,

    50.1, 49.9, 50.8, 50.4, 48.9, 50.6, 50.0, 49.7, 50.2, 49.5,

    50.1, 50.3, 50.4, 50.5, 50.0, 50.7, 49.3, 49.8, 50.2, 50.9,

    50.3, 50.4, 50.0, 49.7, 50.5, 49.9]

Hypothesized population mean

mu_0 = 50

Sample statistics

```
sample_mean = np.mean(sample_data)

sample_std = np.std(sample_data, ddof=1)  # sample standard deviation

n = len(sample_data)


# Z-test statistic

z_stat = (sample_mean - mu_0) / (sample_std / np.sqrt(n))


# Two-tailed p-value

p_value = 2 * (1 - norm.cdf(abs(z_stat)))


print("Sample Mean:", round(sample_mean, 3))

print("Sample Standard Deviation:", round(sample_std, 3))

print("Z-statistic:", round(z_stat, 3))

print("p-value:", round(p_value, 4))


# Decision at α = 0.05

alpha = 0.05

if p_value < alpha:

    print("Reject H0: The sample mean is significantly different from 50.")

else:

    print("Fail to Reject H0: No significant difference from 50.")
```

Explanation of Steps

1. **Null hypothesis (H_0):** $\mu = 50$
2. **Alternative hypothesis (H_1):** $\mu \neq 50$

3. Compute sample mean and sample standard deviation.

4. Calculate Z-statistic:

$$Z = \frac{\bar{X} - \mu_0}{s / \sqrt{n}}$$

5. Compute **p-value** using standard normal distribution.

6. Compare p-value with α = 0.05 → make a decision.

Interpretation (Expected Output)

Suppose the calculations give:

- Sample Mean ≈ **50.05**
- Z-statistic ≈ **0.46**
- p-value ≈ **0.64**

Since **p-value > 0.05**, we **fail to reject H_0**.
That means: **There is no significant evidence that the sample mean differs from 50.**

Question 8: Write a Python script to simulate data from a normal distribution and calculate the 95% confidence interval for its mean. Plot the data using Matplotlib.

Ans.

```
import numpy as np

import matplotlib.pyplot as plt

from scipy import stats


# Step 1: Simulate data from a normal distribution

np.random.seed(42)  # for reproducibility

mu = 50      # true mean

sigma = 5    # true standard deviation

n = 100      # sample size
```

```
data = np.random.normal(mu, sigma, n)


# Step 2: Calculate sample mean and standard error

sample_mean = np.mean(data)

sample_std = np.std(data, ddof=1)

std_error = sample_std / np.sqrt(n)


# Step 3: 95% confidence interval (using t-distribution)

confidence = 0.95

df = n - 1

t_crit = stats.t.ppf((1 + confidence) / 2, df)  # two-tailed critical value

margin_of_error = t_crit * std_error

ci_lower = sample_mean - margin_of_error

ci_upper = sample_mean + margin_of_error


print("Sample Mean:", round(sample_mean, 3))

print("95% Confidence Interval:", (round(ci_lower, 3), round(ci_upper, 3)))


# Step 4: Plot histogram of the data

plt.hist(data, bins=15, edgecolor='black', alpha=0.7, density=True)

plt.axvline(sample_mean, color='red', linestyle='--', label=f"Mean = 
{sample_mean:.2f}")

plt.axvline(ci_lower, color='green', linestyle='--', label=f"95% CI Lower = 
{ci_lower:.2f}")

plt.axvline(ci_upper, color='blue', linestyle='--', label=f"95% CI Upper = {ci_upper:.2f}")


plt.title("Normal Distribution with 95% Confidence Interval")
```

plt.xlabel("Value")

plt.ylabel("Density")

plt.legend()

plt.show()

Explanation

1. Simulate data → `np.random.normal(mu, sigma, n)`
   - mean = 50, std dev = 5, n = 100 samples
2. Compute confidence interval
   CI=X¯±tα/2,df×snCI = \bar{X} \pm t_{\alpha/2, df} \times \frac{s}{\sqrt{n}}CI=X¯±tα/2,df×ns
   where sss = sample standard deviation.
3. Plot
   - Histogram of data
   - Red line = sample mean
   - Green & blue lines = CI boundaries

Interpretation:

- The histogram shows sample distribution.
- The 95% CI indicates the range in which the true mean (50) is expected to lie with 95% confidence.

Question 9: Write a Python function to calculate the Z-scores from a dataset and visualize the standardized data using a histogram. Explain what the Z-scores represent in terms of standard deviations from the mean

Ans. import numpy as np

import matplotlib.pyplot as plt

```
def calculate_zscores(data):
    """
    Function to calculate Z-scores and plot histogram
    """
    mean = np.mean(data)
    std = np.std(data, ddof=1)  # sample standard deviation

    # Calculate Z-scores
    z_scores = (data - mean) / std

    # Plot histogram
    plt.hist(z_scores, bins=15, edgecolor='black', alpha=0.7, density=True)
    plt.axvline(0, color='red', linestyle='--', label="Mean (Z=0)")
    plt.title("Histogram of Standardized Data (Z-scores)")
    plt.xlabel("Z-score")
    plt.ylabel("Density")
    plt.legend()
    plt.show()

    return z_scores

# Example usage
data = [49.1, 50.2, 51.0, 48.7, 50.5, 49.8, 50.3, 50.7, 50.2, 49.6,
        50.1, 49.9, 50.8, 50.4, 48.9, 50.6, 50.0, 49.7, 50.2, 49.5,
```

Explanation of Z-scores

- **Definition:**
  A Z-score tells us how many standard deviations a data point is away from the mean.

- **Formula:**
  Z=X−X‾sZ = \frac{X - \bar{X}}{s}Z=sX−X‾
  where

  - **XXX = data point**
  - **X‾\bar{X}X‾ = sample mean**
  - **sss = sample standard deviation**

Interpretation

- **Z = 0 → The value is exactly at the mean.**
- **Z = +1 → 1 standard deviation above the mean.**
- **Z = -2 → 2 standard deviations below the mean.**
- **Helps standardize data (mean = 0, std dev = 1) for comparisons.**

.