Q1. What is hypothesis testing in statistics?

Hypothesis testing is a statistical procedure used to make decisions about population parameters based on sample data. It begins with a **null hypothesis (H_0)**, which assumes no effect, difference, or relationship exists. An **alternative hypothesis (H_1)** represents the claim being tested. Using sample information, a test statistic is calculated and compared to a probability distribution to determine whether the observed data could reasonably occur under H_0. If the evidence is strong, H_0 is rejected in favor of H_1. Hypothesis testing is widely used in research, business, and science to validate claims, compare groups, and estimate population behavior.

Q2. What is the null hypothesis, and how does it differ from the alternative hypothesis?

The **null hypothesis (H_0)** is a baseline assumption that there is no effect, no change, or no difference in a population. It serves as the default statement that researchers attempt to test. The **alternative hypothesis (H_1 or Ha)** represents the opposite claim, suggesting that an effect or difference does exist. Hypothesis testing evaluates whether sample evidence is strong enough to reject H_0. For example, if a manufacturer claims a product weighs 50g, H_0 states $\mu = 50$, while H_1 may state $\mu \neq 50$. The goal is not to prove H_0 true but to determine whether data contradict it.

Q3. Explain the significance level in hypothesis testing and its role in deciding the outcome of a test.

The **significance level (α)** is the probability threshold used to decide whether to reject the null hypothesis. It represents the risk of making a **Type I error**, which occurs when a true H_0 is rejected. Common significance levels are **0.05**, **0.01**, and **0.10**. During hypothesis testing, the **p-value** is compared to α. If the p-value is smaller, the result is statistically significant, and H_0 is rejected. If the p-value is larger, H_0 is not rejected. The significance level ensures consistency and helps researchers control the likelihood of false positives when interpreting statistical evidence.

Q4. What are Type I and Type II errors? Give examples of each.

A **Type I error** occurs when the null hypothesis is true but is incorrectly rejected. It is known as a **false positive**, and its probability is equal to the significance level α. For example, concluding that a new medicine works when it actually has no effect is a Type I error. A **Type II error** happens when the null hypothesis is false, but the test fails to reject it, resulting in a **false negative**. For instance, concluding that a medicine does not work when it truly is effective. Type II error probability is represented by β, and reducing such errors increases statistical power.

Q5. What is the difference between a Z-test and a T-test? Explain when to use each.

A **Z-test** is used when the population standard deviation is known or when the sample size is large (typically above 30). It relies on the standard normal distribution. A **T-test** is used when the population standard deviation is unknown and the sample size is small. It uses the t-distribution, which has thicker tails to account for increased variability in small samples. Both tests compare a sample mean to a population mean, but the choice depends on sample size and whether σ is known. In most practical situations, the T-test is preferred because population variance is rarely available.

Q6. Write a Python program to generate a binomial distribution with n=10 and p=0.5, then plot its histogram.

A binomial distribution models the number of successes in a fixed number of independent trials, each having the same probability of success. In this case, we generate random values using **n = 10** and **p = 0.5**, meaning each trial has a 50% chance of success. Using Python's NumPy library, we can generate many such outcomes and plot a histogram to visualize their distribution. As the number of samples increases, the histogram generally forms a symmetric shape centered around **np = 5**. This simulation helps understand theoretical distributions and observe how trial probability influences outcomes.

Python Code

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

data = np.random.binomial(10, 0.5, 1000)
```

```
plt.hist(data)
plt.show()
```

Q7. 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]

To perform hypothesis testing using Z-statistics, we compare the sample mean to a hypothesized population mean. First, we compute the sample mean and standard deviation from the data provided. We assume the null hypothesis H_0: $\mu = 50$. The Z-statistic measures how many standard errors the sample mean deviates from 50. Using the standard normal distribution, we calculate a p-value. If the p-value is below the significance level, we reject H_0. In this dataset, the p-value is relatively high, meaning the sample mean does not significantly differ from 50, so we fail to reject H_0.

Python Code

```
import numpy as np
from scipy import stats

sample = np.array([...])
z = (sample.mean() - 50) / (sample.std(ddof=1) / np.sqrt(len(sample)))
p = 2 * (1 - stats.norm.cdf(abs(z)))
print(z, p)
```

Q8. 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.

Simulating data from a normal distribution allows us to observe how real-world measurements might vary around a mean. By generating values from a distribution with mean 50 and standard deviation 5, we can visualize the data using a histogram, which typically forms a bell-shaped curve. To estimate the population mean, we calculate a **95% confidence interval**, which gives a range of plausible values for the true mean. The interval is computed as: mean ± 1.96 × (standard error). This range reflects the uncertainty due to sampling and becomes narrower with larger sample sizes.

Python Code

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

data = np.random.normal(50, 5, 1000)
mean = data.mean()
ci = (mean - 1.96*(data.std(ddof=1)/np.sqrt(1000)),
      mean + 1.96*(data.std(ddof=1)/np.sqrt(1000)))
print(ci)
plt.hist(data)
plt.show()
```

Q9. 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.

Z-scores measure how far each data point lies from the mean in units of standard deviation. A Z-score of 0 means the value is exactly at the mean, while +1 or –1 indicates the value is one standard deviation above or below the mean. Standardizing data transforms it into a distribution with mean 0 and standard deviation 1, making comparison easier across different scales. Plotting a histogram of these Z-scores typically reveals a standard normal shape. Z-scores are useful for detecting outliers, normalizing datasets, and comparing values measured in different units or contexts.

Python Code

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

def zscores(x): return (x - np.mean(x)) / np.std(x, ddof=1)
```

```
zs = zscores(data)
plt.hist(zs)
plt.show()
```