STATISTICS ADVANCE – 2 ASSIGNMENT

Question 1: What is hypothesis testing in statistics?

Answer :

Hypothesis testing in statistics is a method used to make decisions or inferences about a population based on sample data. It involves formulating two competing statements: the **null hypothesis (H_o)**, which represents a default or no-effect scenario, and the **alternative hypothesis (H_1)**, which suggests a significant effect or difference. The process involves collecting data, analyzing it using a statistical test, and determining whether there is enough evidence to reject the null hypothesis in favor of the alternative, based on a chosen significance level (commonly 0.05). This helps researchers make objective conclusions about their data.

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

Ansewer :

The **null hypothesis (H_o)** is a statement that assumes there is no effect, difference, or relationship between variables—it represents the status quo or a baseline assumption. In contrast, the **alternative hypothesis (H_1 or Ha)** suggests that there *is* an effect, difference, or relationship. The goal of hypothesis testing is to use data to determine whether there is enough evidence to reject the null hypothesis in favor of the alternative. Essentially, the null is assumed true until proven otherwise, while the alternative is accepted only if the data provides strong enough evidence.

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

Answer :

The **significance level** (denoted as **α**) in hypothesis testing is the threshold used to decide whether to reject the null hypothesis. It represents the probability of making a **Type I error**, which is rejecting the null hypothesis when it is actually true. Commonly set at 0.05 (or 5%), it means there's a 5% risk of concluding that an effect exists when it doesn't. If the **p-value** from the test is less than or equal to the significance level, the null hypothesis is rejected. Thus, the significance level helps determine how strong the evidence must be to consider the test results statistically significant.

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

Answer :

In hypothesis testing, a **Type I error** occurs when the **null hypothesis is wrongly rejected**, even though it is true. For example, concluding a new drug works better than the old one when it actually doesn’t. A **Type II error** happens when the **null hypothesis is wrongly accepted**, meaning we fail to detect a real effect. For instance, concluding a new teaching method has no impact on student performance when it actually does. Type I error is also known as a "false positive," while Type II is a "false negative." Both errors reflect the risks involved in statistical decision-making.

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

Answer :

A **Z-test** and a **T-test** are both used to compare sample data to a population or between groups, but they differ in when they are applied. A **Z-test** is used when the population **standard deviation is known** and the **sample size is large** (typically n > 30), assuming a normal distribution. A **T-test**, on the other hand, is used when the population standard deviation is **unknown** and the **sample size is small** (n ≤ 30). The T-test accounts for more uncertainty by using the t-distribution, which is wider than the normal distribution, especially with smaller samples.

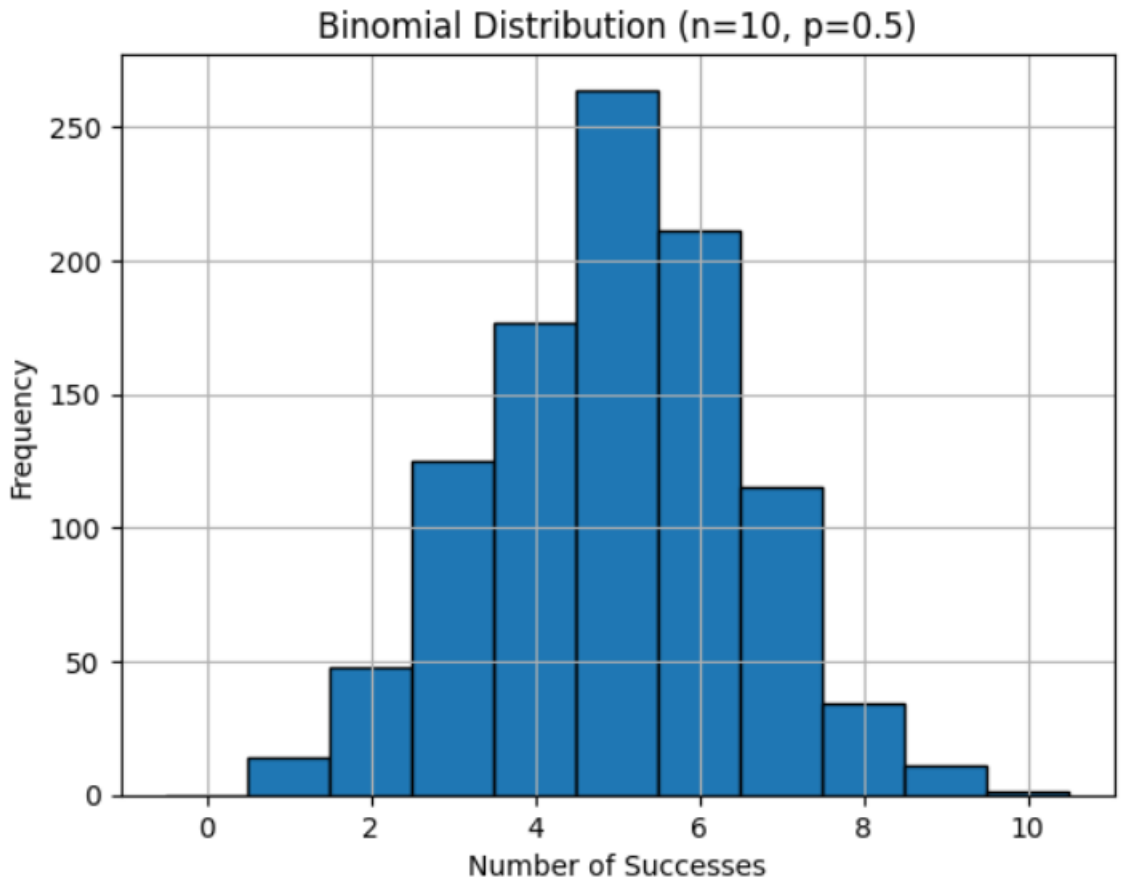
Question 6: Write a Python program to generate a binomial distribution with n=10 and p=0.5, then plot its histogram.

Hint: Generate random number using random function.

Answer :

CODE :

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


# Plot histogram

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

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

plt.xlabel('Number of Successes')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()
```



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]

Answer :

CODE :

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]

# Known population standard deviation
sigma = 0.5
# Hypothesized population mean
mu = 50
# Sample mean and size
sample_mean = np.mean(sample_data)
n = len(sample_data)
# Calculate Z-statistic
z_stat = (sample_mean - mu) / (sigma / np.sqrt(n))
# Calculate p-value (two-tailed)
p_value = 2 * (1 - norm.cdf(abs(z_stat)))

print(f"Sample Mean = {sample_mean:.4f}")
print(f"Z-Statistic = {z_stat:.4f}")
print(f"P-Value = {p_value:.4f}")

# Interpret the result
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference.")
else:
    print("Fail to reject the null hypothesis: No significant difference.")
```

OUTOUT :

Sample Mean = 50.0889

Z-Statistic = 1.0667

P-Value = 0.2861

Fail to reject the null hypothesis: No significant difference.

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.

Answer :

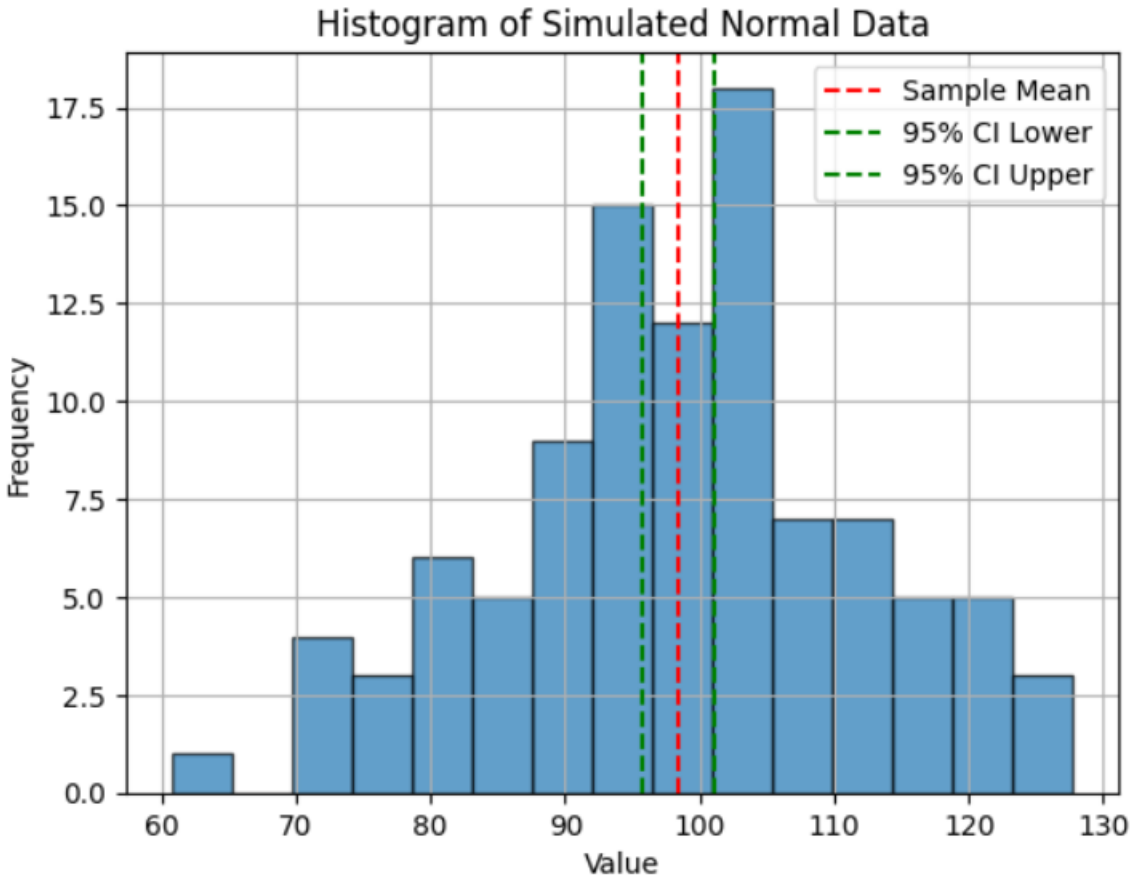
CODE :

```
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 = 100     # true mean

sigma = 15   # true standard deviation

n = 100      # sample size


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


# Step 2: Calculate the 95% confidence interval for the mean

sample_mean = np.mean(data)

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

confidence = 0.95

alpha = 1 - confidence


# t-score for 95% confidence with df = n - 1

t_critical = stats.t.ppf(1 - alpha/2, df=n-1)
```

```
# Margin of error
margin_error = t_critical * (sample_std / np.sqrt(n))

# Confidence interval
ci_lower = sample_mean - margin_error
ci_upper = sample_mean + margin_error

# Step 3: Output results
print(f"Sample Mean = {sample_mean:.2f}")
print(f"95% Confidence Interval = ({ci_lower:.2f}, {ci_upper:.2f})")

# Step 4: Plot the data
plt.hist(data, bins=15, edgecolor='black', alpha=0.7)
plt.axvline(sample_mean, color='red', linestyle='--', label='Sample Mean')
plt.axvline(ci_lower, color='green', linestyle='--', label='95% CI Lower')
plt.axvline(ci_upper, color='green', linestyle='--', label='95% CI Upper')
plt.title('Histogram of Simulated Normal Data')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT :

```
Sample Mean = 98.44
95% Confidence Interval = (95.74, 101.15)
```



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.

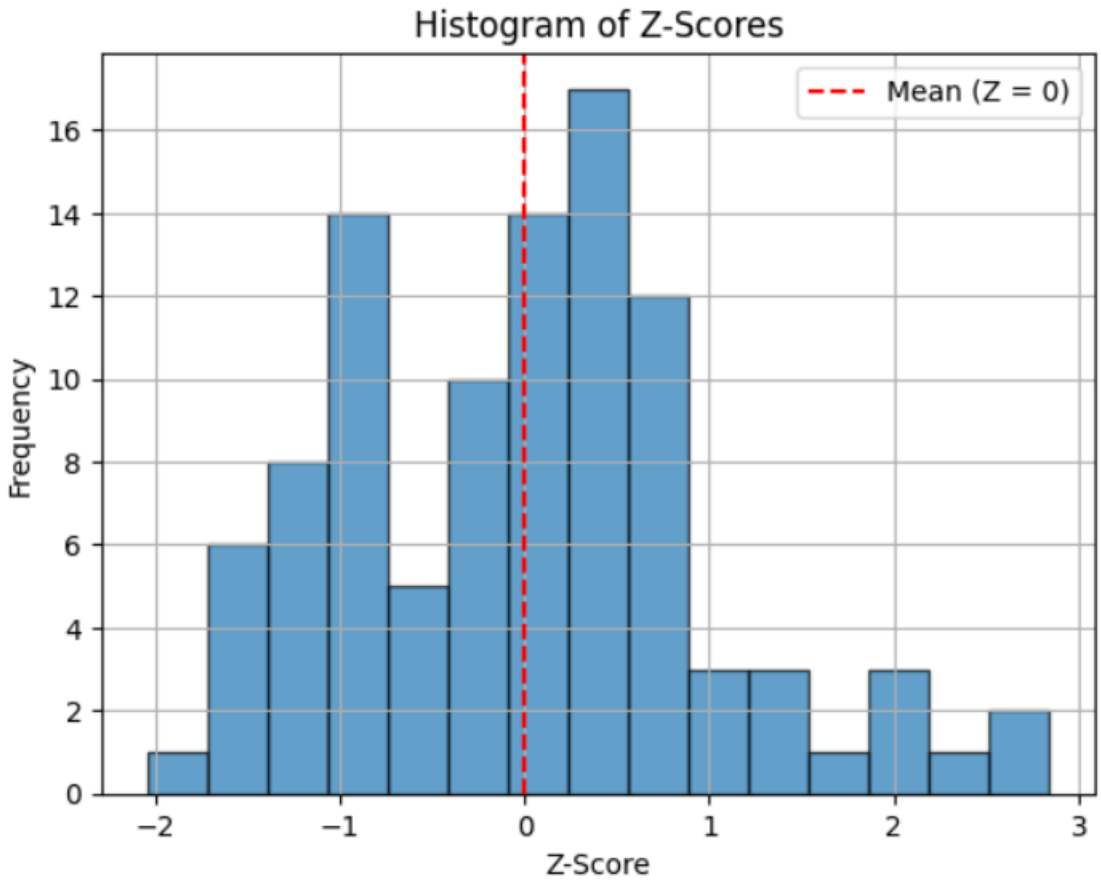
Answer :

CODE :

```
import numpy as np

import matplotlib.pyplot as plt


def calculate_z_scores(data):

    # Calculate mean and standard deviation

    mean = np.mean(data)

    std = np.std(data)
```

```
    # Calculate Z-scores
    z_scores = [(x - mean) / std for x in data]

    # Plot histogram of Z-scores
    plt.hist(z_scores, bins=15, edgecolor='black', alpha=0.7)
    plt.title('Histogram of Z-Scores')
    plt.xlabel('Z-Score')
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.axvline(0, color='red', linestyle='--', label='Mean (Z = 0)')
    plt.legend()
    plt.show()

    return z_scores

# Example usage with random data
data = np.random.normal(100, 15, 100)
z_scores = calculate_z_scores(data)

# Optional: print a few Z-scores
print("First 5 Z-scores:", np.round(z_scores[:5], 2))
```

OUTPUT :




```
First 5 Z-scores: [-1.52 -0.47 -0.38 -0.87 -0.19]
```