Statistics Advanced - 2| Assignment

Question 1: What is hypothesis testing in statistics?

Answer: Hypothesis testing is a statistical technique that evaluates two competing statements (hypotheses) about a population using data from a sample. The goal is to assess whether the evidence supports rejecting one of the hypotheses in favor of the other.

Null Hypothesis (H_0): The default assumption that there is no effect or no difference. Example: "The average website visits per day is 50."

Alternative Hypothesis (H_1): The statement that contradicts the null hypothesis. Example: "The average website visits per day is not 50"

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

Null Hypothesis (H_0)

- **Definition:** The null hypothesis is a default assumption that there is *no effect*, *no difference*, or *no relationship* between variables.
- **Purpose:** It serves as the starting point for statistical testing.
- **Example:** If a company claims their battery lasts 10 hours, H_0 might be: “The average battery life is 10 hours.”

Alternative Hypothesis (H_1 or Ha)

- **Definition:** The alternative hypothesis contradicts the null hypothesis. It suggests that there *is* an effect, a difference, or a relationship.
- **Purpose:** It represents what the researcher aims to support.
- **Example:** Continuing the battery example, H_1 might be: “The average battery life is not 10 hours.”

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

Answer: The **significance level (α)** is the threshold used in hypothesis testing to decide whether to reject the null hypothesis. It represents the probability of making a **Type I error**—rejecting a true null hypothesis. Common values are **0.05**, **0.01**, or **0.10**. If the **p-value ≤ α**, we reject the null hypothesis, indicating the result is statistically significant.

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

Answer:

Type I Error (False Positive)

- **Definition:** Occurs when the null hypothesis (H_0) is true, but you mistakenly reject it.
- **Symbol:** α (alpha), which is the significance level.
- **Example:** A medical test incorrectly indicates a patient has a disease when they actually don't.

Type II Error (False Negative)

- **Definition:** Happens when the null hypothesis is false, but you fail to reject it.
- **Symbol:** β (beta), which relates to the test's power.
- **Example:** A medical test fails to detect a disease in a patient who actually has it.

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

Answer:

Feature	Z-Test	T-Test
Sample Size	Large (typically > 30)	Small (typically ≤ 30)
Population Std. Dev.	Known	Unknown
Distribution Used	Standard Normal (Z-distribution)	Student's t-distribution
Use Case	Comparing means when variance is known	Comparing means when variance is unknown

Use Z-test: For large samples with known population variance.

Use T-test: For small samples or when population variance is unknown.

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

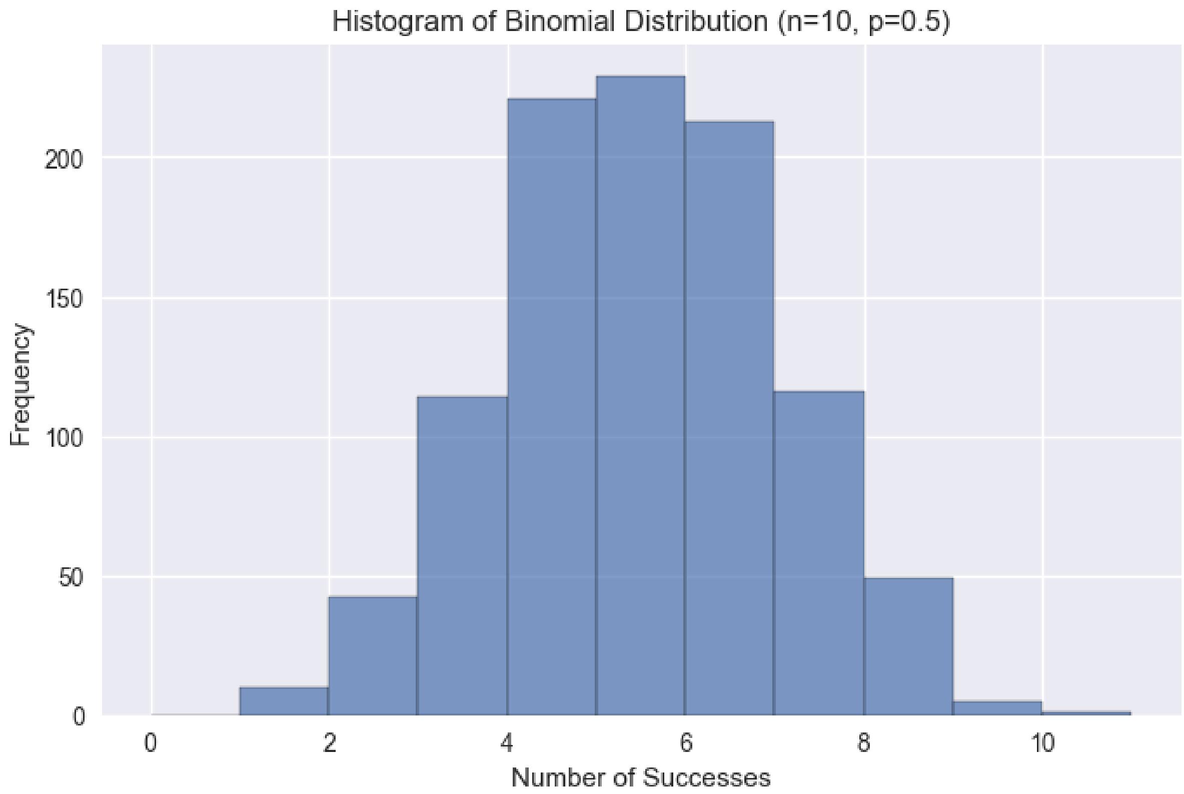
Answer:

- **Z-test** is used when the **sample size is large ($n > 30$)** and the **population standard deviation is known**.
- **T-test** is used when the **sample size is small ($n \leq 30$)** and the **population standard deviation is unknown**.
- • **Z-test**: Large sample, known variance — e.g., testing population mean with known σ.
- • **T-test**: Small sample, unknown variance — e.g., comparing means from small groups or paired samples.

Question 6: Write a Python program to generate a binomial distribution with n=10 and p=0.5, then plot its histogram. (Include your Python code and output in the code box below.) Hint: Generate random number using random function.

Answer:

```
import numpy as np

import matplotlib.pyplot as plt



# Generate 1000 samples from a binomial distribution with n=10 and p=0.5

n = 10

p = 0.5

samples = np.random.binomial(n=n, p=p, size=1000)



# Plot histogram

plt.style.use('seaborn-v0_8')

plt.figure(figsize=(8, 5))

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

plt.title('Histogram of 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] (Include your Python code and output in the code box below.)

Answer:

```
import numpy as np

from scipy.stats import norm



# Given 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]
```

```
# Parameters
mu = 50          # Population mean
sigma = 0.5      # Population standard deviation
alpha = 0.05     # Significance level

# Step 1: Calculate sample mean
sample_mean = np.mean(sample_data)
n = len(sample_data)

# Step 2: Compute Z-statistic
z_stat = (sample_mean - mu) / (sigma / np.sqrt(n))

# Step 3: Calculate two-tailed p-value
p_value = 2 * (1 - norm.cdf(abs(z_stat)))

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

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

Output :

Sample Mean: 50.0889

Z-Statistic: 1.0667

P-Value: 0.2861

Conclusion: Fail to reject the null hypothesis.

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. (Include your Python code and output in the code box below.)

Answer: import numpy as np

import scipy.stats as stats

import matplotlib.pyplot as plt

Simulate data

data = np.random.normal(loc=100, scale=15, size=1000)

Calculate sample mean and standard error

sample_mean = np.mean(data)

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

standard_error = sample_std / np.sqrt(len(data))

Compute 95% confidence interval

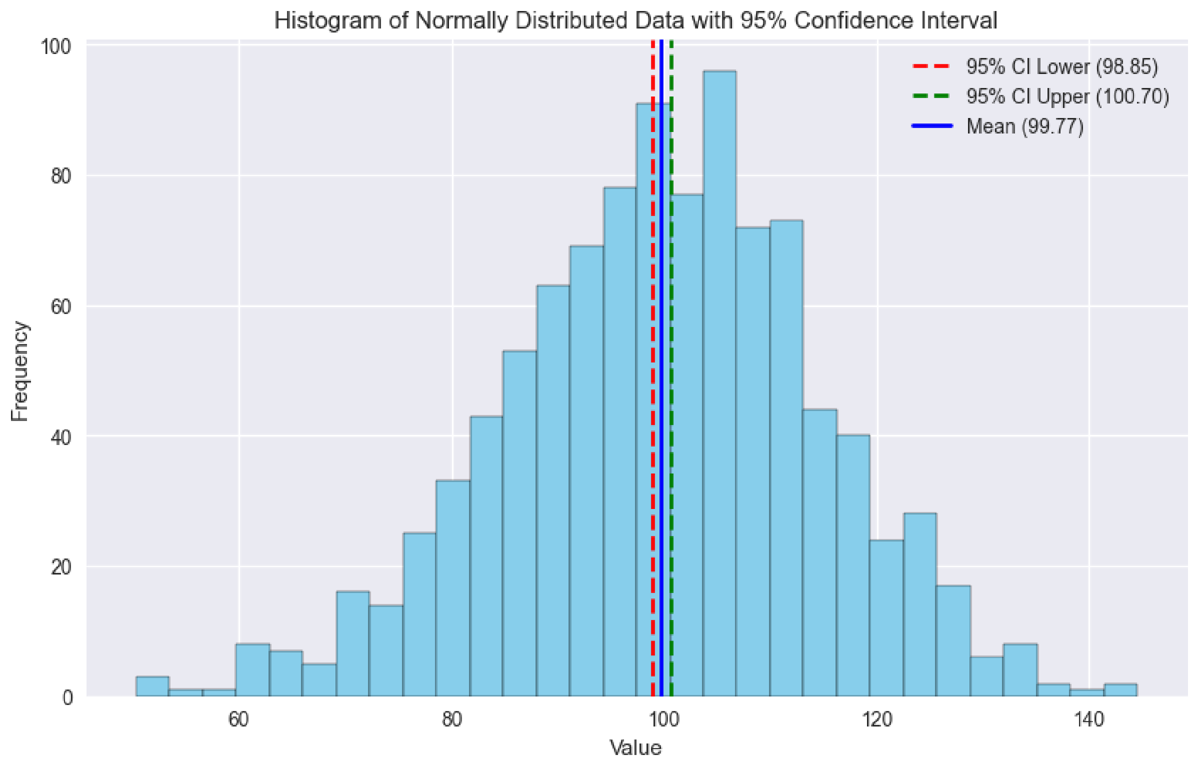
confidence_level = 0.95

ci_low, ci_high = stats.norm.interval(confidence_level, loc=sample_mean, scale=standard_error)

Plot histogram

```
plt.style.use('seaborn-v0_8')

plt.figure(figsize=(10, 6))

plt.hist(data, bins=30, color='skyblue', edgecolor='black')

plt.axvline(ci_low, color='red', linestyle='dashed', linewidth=2, label=f'95% CI Lower
({ci_low:.2f})')

plt.axvline(ci_high, color='green', linestyle='dashed', linewidth=2, label=f'95% CI Upper
({ci_high:.2f})')

plt.axvline(sample_mean, color='blue', linestyle='solid', linewidth=2, label=f'Mean
({sample_mean:.2f})')



# Add labels and legend

plt.title('Histogram of Normally Distributed Data with 95% Confidence Interval')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.legend()

plt.grid(True)

plt.show()
```

Output:



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. (Include your Python code and output in the code box below.)

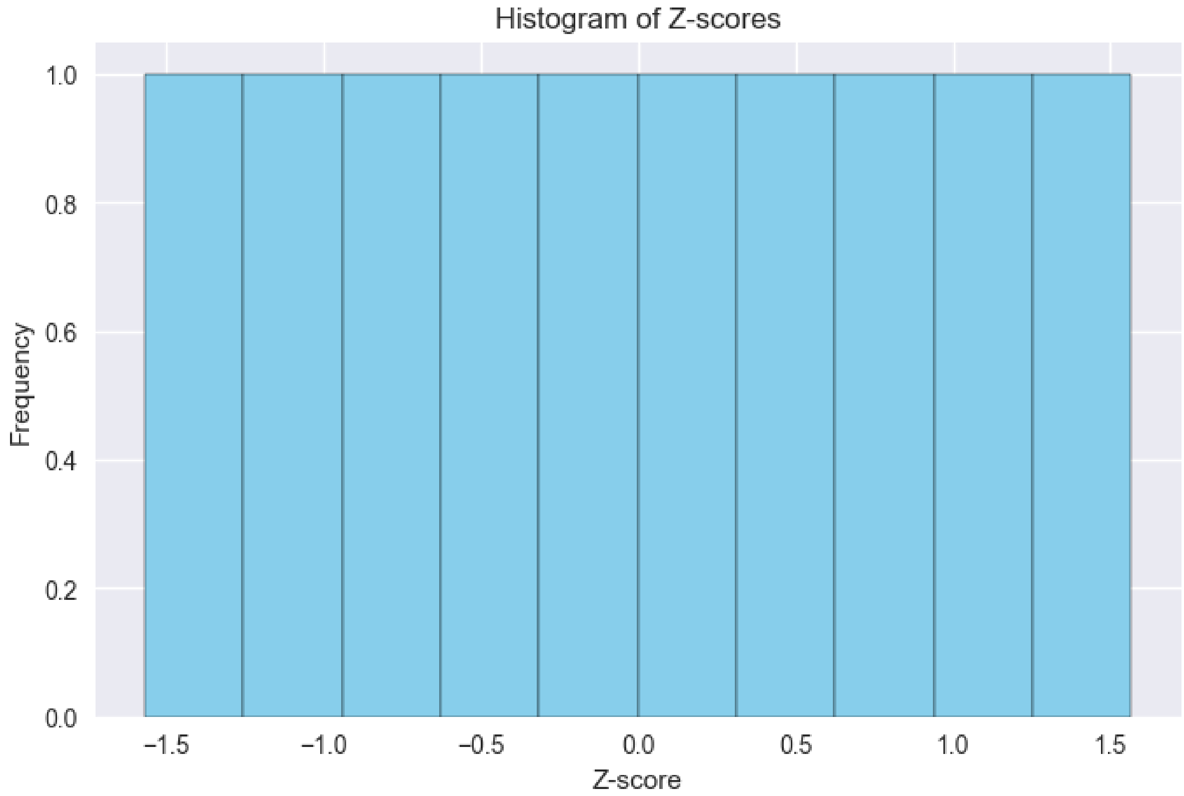
Answer:

- A **Z-score** tells you how far a value is from the mean in terms of standard deviations.
- **Z = 0** → value equals the mean.
- **Z > 0** → value is above the mean.
- **Z < 0** → value is below the mean.

```
import numpy as np

import matplotlib.pyplot as plt



def calculate_z_scores(data):

    mean = np.mean(data)

    std = np.std(data)

    z_scores = [(x - mean) / std for x in data]

    return z_scores
```

```
# Sample dataset
data = [55, 60, 65, 70, 75, 80, 85, 90, 95, 100]

# Calculate Z-scores
z_scores = calculate_z_scores(data)

# Plot histogram of Z-scores
plt.style.use('seaborn-v0_8')
plt.figure(figsize=(8, 5))
plt.hist(z_scores, bins=10, edgecolor='black', color='skyblue')
plt.title('Histogram of Z-scores')
plt.xlabel('Z-score')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

output:



Histogram of Z-scores
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Z-score