THEORY QUESTIONS

Question 1: What is hypothesis testing in statistics?

Answer:
Hypothesis testing is a statistical method used to make decisions about a population based on sample data. It involves formulating two competing hypotheses, the null and the alternative, and using sample data to determine which one is more likely to be true. This process helps to determine if an observed effect in a sample is statistically significant or if it could have occurred by chance.

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

Answer:

- **Null Hypothesis (H_0):**

  The null hypothesis is a statement of no effect or no difference. It's the default assumption that there's no relationship between variables or that a treatment has no effect. The goal of a hypothesis test is to see if we have enough evidence to reject this hypothesis. For example, if you're testing a new drug, the null hypothesis would be that the drug has no effect.

- **Alternative Hypothesis (H_a or H_1):**

  The alternative hypothesis is the opposite of the null hypothesis. It's the statement that a difference or effect exists. It's what you hope to prove. Following the drug example, the alternative hypothesis would be that the drug does have an effect.

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

Answer:

The significance level (

alpha) is the probability of rejecting the null hypothesis when it is actually true. It represents the threshold for deciding whether the results of a statistical test are statistically significant. A common value for alpha is 0.05, meaning there's a 5% chance of incorrectly rejecting the null hypothesis. The significance level plays a crucial role in deciding the outcome of a test:

- If the **p-value** (the probability of observing your data or more extreme data, given that the null hypothesis is true) is **less than or equal to**

alpha, you **reject** the null hypothesi

- If the **p-value** is **greater than**

alpha, you **fail to reject** the null hypothesis. This means there isn't enough evidence to conclude the effect is real, and the observed results could be due to chance.

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

Answer:

- **Type I Error (**

  alpha**)**: A Type I error occurs when you **reject a true null hypothesis**. This is often called a "false positive." It means you conclude there is an effect or a difference when there isn't one. An example would be a medical test incorrectly indicating a patient has a disease when they are healthy.

- **Type II Error (**

  beta**)**: A Type II error occurs when you **fail to reject a false null hypothesis**. This is often called a "false negative." It means you fail to detect an effect or a difference that actually exists. An example would be a medical test failing to detect a disease in a patient who is actually sick.

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

Answer:

The key difference between a Z-test and a T-test lies in their assumptions about the population **standard deviation.**

- **Z-test:** Use a Z-test when the **population standard deviation (**

  sigma) **is known.** It's also applicable when the sample size is large (typically n

  geq30), as the sample standard deviation then closely approximates the population standard deviation.

- **T-test:** Use a T-test when the **population standard deviation is unknown**, which is the

more common scenario. The test uses the sample standard deviation (s) to estimate the

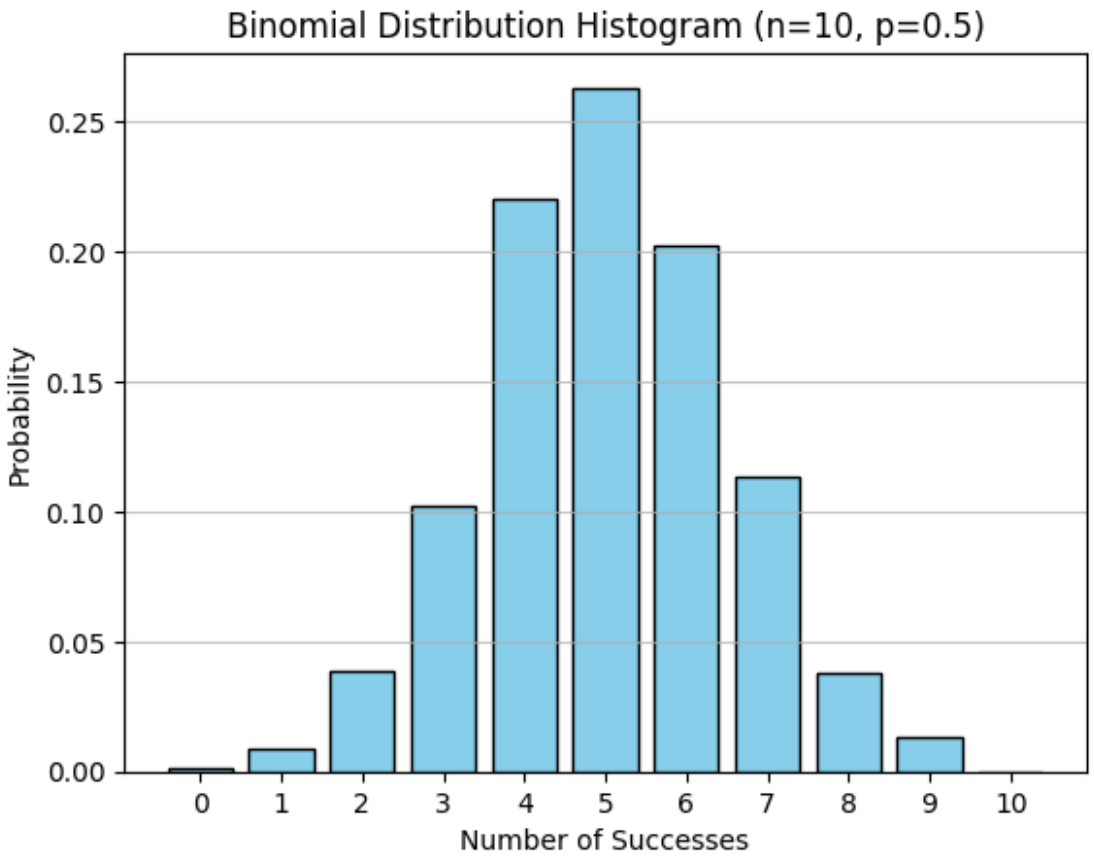
population standard deviation. The t-distribution accounts for the additional uncertainty that comes from estimating the standard deviation from the sample.

PRACTICAL QUESTIONS

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

Answer:

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

# Parameters for the binomial distribution
n = 10  # Number of trials
p = 0.5 # Probability of success

# Generate random numbers from a binomial distribution
# The 'size' parameter determines how many random numbers to generate
num_samples = 1000
random_numbers = np.random.binomial(n, p, size=num_samples)

# Create a histogram
plt.hist(random_numbers, bins=np.arange(n + 2) - 0.5, density=True,
rwidth=0.8, color='skyblue', edgecolor='black')
plt.title(f'Binomial Distribution Histogram (n={n}, p={p})')
plt.xlabel('Number of Successes')
plt.ylabel('Probability')
plt.xticks(range(n + 1))
plt.grid(axis='y', alpha=0.75)
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:

```
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]

# Parameters
population_mean = 50.0
population_std = 0.5  # known
```

```
alpha = 0.05

# Sample statistics
sample_mean = np.mean(sample_data)
sample_size = len(sample_data)
standard_error = population_std / np.sqrt(sample_size)

# Z statistic
z_stat = (sample_mean - population_mean) / standard_error

# p-value for two-tailed test
p_value = 2 * (1 - norm.cdf(abs(z_stat)))

# Print results
print(f"Sample Mean: {sample_mean:.4f}")
print(f"Z-statistic: {z_stat:.4f}")
print(f"P-value: {p_value:.4f}")

# Interpretation
if p_value < alpha:
    print("Reject the null hypothesis: significant difference from the
population mean.")
else:
    print("Fail to reject the null hypothesis: no significant
difference from the population mean.")
```

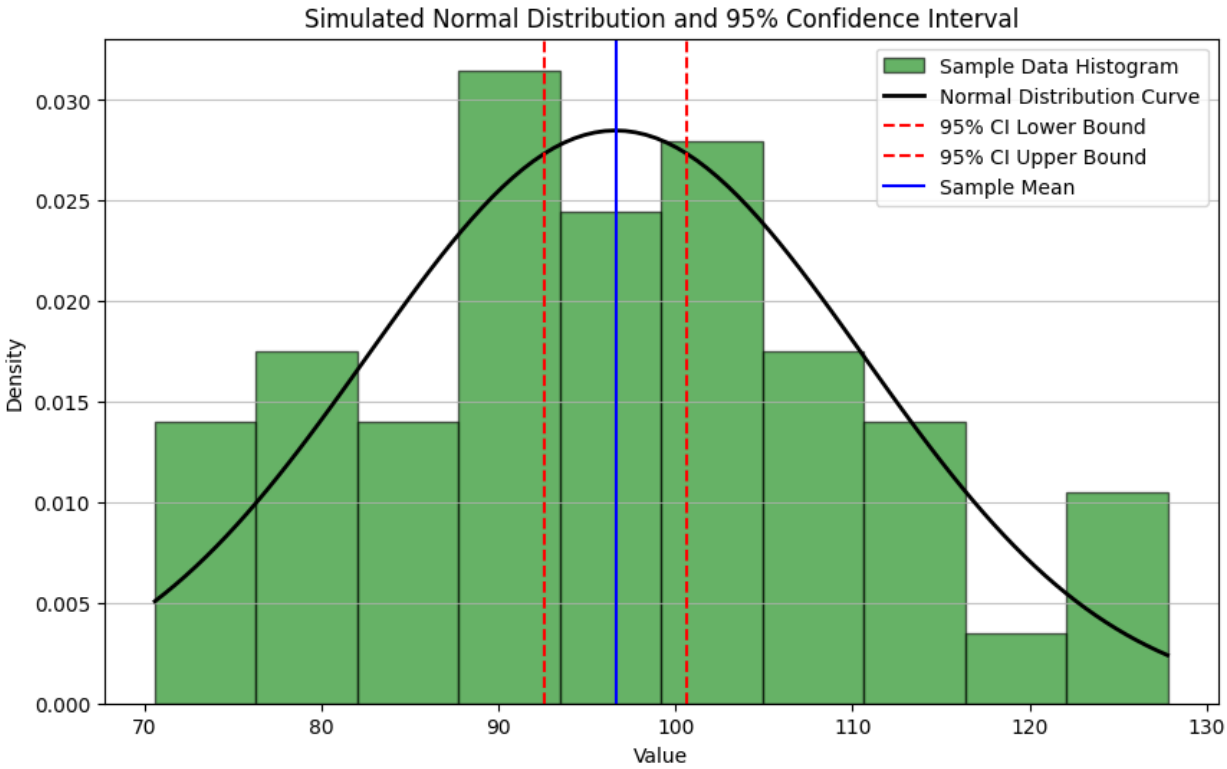
```
Sample Mean: 50.0889
Z-statistic: 1.0667
P-value: 0.2861
Fail to reject the null hypothesis: no significant difference from the
population mean.
```

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:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# Simulate data from a normal distribution
mu = 100       # Population mean
sigma = 15     # Population standard deviation
sample_size = 50
np.random.seed(42)  # for reproducibility
sample_data = np.random.normal(mu, sigma, sample_size)
```

```
# Calculate the sample mean and standard error
sample_mean = np.mean(sample_data)
standard_error = stats.sem(sample_data)

# Calculate the 95% confidence interval
confidence_level = 0.95
degrees_freedom = sample_size - 1
t_critical = stats.t.ppf((1 + confidence_level) / 2, degrees_freedom)
margin_of_error = t_critical * standard_error
confidence_interval = (sample_mean - margin_of_error, sample_mean +
margin_of_error)

print(f"Sample Mean: {sample_mean:.2f}")
print(f"95% Confidence Interval: ({confidence_interval[0]:.2f},
{confidence_interval[1]:.2f})")

# Plot the data using Matplotlib
plt.figure(figsize=(10, 6))
plt.hist(sample_data, bins=10, density=True, alpha=0.6, color='g',
edgecolor='black', label='Sample Data Histogram')

# Plot the normal distribution curve
x = np.linspace(min(sample_data), max(sample_data), 100)
pdf = stats.norm.pdf(x, sample_mean, np.std(sample_data, ddof=1))
plt.plot(x, pdf, 'k-', linewidth=2, label='Normal Distribution Curve')

# Plot the confidence interval
plt.axvline(x=confidence_interval[0], color='r', linestyle='--',
label=f'95% CI Lower Bound')
plt.axvline(x=confidence_interval[1], color='r', linestyle='--',
label=f'95% CI Upper Bound')
plt.axvline(x=sample_mean, color='b', linestyle='-', label=f'Sample
Mean')

plt.title('Simulated Normal Distribution and 95% Confidence Interval')
plt.xlabel('Value')
plt.ylabel('Density')
plt.legend()
plt.grid(axis='y', alpha=0.75)
plt.show()

Sample Mean: 96.62
95% Confidence Interval: (92.64, 100.60)
```



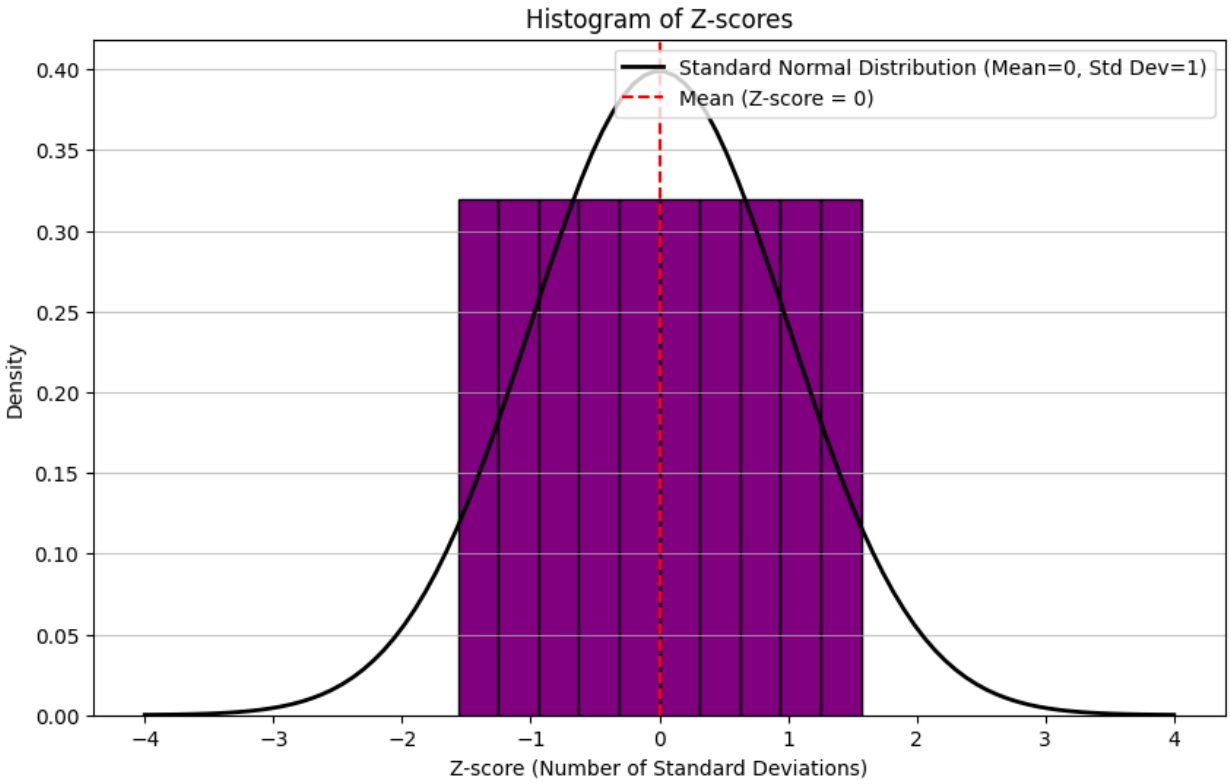
Question 9: Write a Python function to calculate the Z-scores from a dataset and visualize the standardized data using a histogram. Explain what the Z-scores represent in terms of standard deviations from the mean.

Answer:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import zscore

def visualize_zscores(data):
    """
    Calculates z-scores for a dataset and visualizes the standardized
data.

    Args:
        data (array-like): The input dataset.
    """
    # Calculate Z-scores
    z_scores = zscore(data)

    # Explain what Z-scores represent
    print("Z-scores represent the number of standard deviations a data
point is away from the mean.")
```

```
    print("A positive Z-score means the value is above the mean, while
a negative Z-score means it is below the mean.")
    print("A Z-score of 0 indicates the value is equal to the mean.")

    # Visualize the standardized data using a histogram
    plt.figure(figsize=(10, 6))
    plt.hist(z_scores, bins=10, density=True, color='purple',
edgecolor='black')

    # Plot a standard normal distribution curve for comparison
    x = np.linspace(-4, 4, 100)
    pdf = stats.norm.pdf(x, 0, 1)
    plt.plot(x, pdf, 'k-', linewidth=2, label='Standard Normal
Distribution (Mean=0, Std Dev=1)')

    plt.title('Histogram of Z-scores')
    plt.xlabel('Z-score (Number of Standard Deviations)')
    plt.ylabel('Density')
    plt.axvline(0, color='r', linestyle='--', label='Mean (Z-score =
0)')
    plt.legend()
    plt.grid(axis='y', alpha=0.75)
    plt.show()

    return z_scores

# Example usage with a sample dataset
sample_data_q9 = np.array([65, 70, 75, 80, 85, 90, 95, 100, 105, 110])
z_scores_q9 = visualize_zscores(sample_data_q9)
print("\nOriginal Data:", sample_data_q9)
print("Calculated Z-scores:", z_scores_q9)
```

```
Z-scores represent the number of standard deviations a data point is
away from the mean.
A positive Z-score means the value is above the mean, while a negative
Z-score means it is below the mean.
A Z-score of 0 indicates the value is equal to the mean.
```




```
Original Data: [ 65  70  75  80  85  90  95 100 105 110]
Calculated Z-scores: [-1.5666989  -1.21854359 -0.87038828 -0.52223297
-0.17407766  0.17407766
  0.52223297  0.87038828  1.21854359  1.5666989 ]
```