Statistics Advanced - 2| Assignment

Question 1: What is hypothesis testing in statistics?

Answer :

Hypothesis testing in statistics is a formal procedure used to make inferences or draw conclusions about a population based on sample data. It helps determine whether there is enough evidence in a sample to support or reject a specific claim about a population parameter.

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

Answer :

The null hypothesis is a statement that assumes no effect, no difference, or no relationship between variables. It serves as the default or starting assumption in hypothesis testing and is tested directly using sample data.

- It is considered true until there is strong evidence against it.
- Often includes equality (e.g., =, ≥, ≤).

Difference between null alternative hypothesis :

- Null hypothesis has no effect while alternative hypothesis has an effect.
- Purpose of null hypothesis is starting point for testing while purpose of alternative hypothesis is represent the claim being tested.
- Null hypothesis is used mathematical symbols(=, <=, >=) while alternative hypothesis is used mathematical symbols(≠, <, >)

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

Answer :

Significance level :

- The significance level, denoted by α (alpha), is a threshold set before conducting a hypothesis test that defines how much risk you are willing to take in rejecting a true null hypothesis. In other words, it's the probability of making a Type I error.

Role in Deciding the Outcome of a Test :

Once the p-value (probability of observing the data assuming H_0 is true) is calculated from the test, it is compared to the significance level:

Decision Rule:

- If p-value $\leq \alpha \rightarrow$ Reject the null hypothesis (evidence supports H_1)
- If p-value $> \alpha \rightarrow$ Fail to reject the null hypothesis (not enough evidence)

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

Answer :

1. Type I Error (False Positive):

- Definition: Rejecting the null hypothesis (H_0) when it is actually true.
- This means you detect an effect or difference when none actually exists.
- The probability of making a Type I error is the significance level (α).

Example:

Imagine a new drug is being tested:

- **H_0:** The drug has no effect.
- **H_1:** The drug has a positive effect.

If researchers reject H_0 and conclude the drug works, but in reality it doesn't, they've made a Type I error.

2. Type II Error (False Negative):

- Definition: Failing to reject the null hypothesis (H_0) when it is actually false.
- This means you miss a real effect or difference.
- The probability of a Type II error is denoted by $\boldsymbol{\beta}$ (beta).

Example:

Using the same drug test:

- **H_0:** The drug has no effect.
- **H_1:** The drug has a positive effect.

If researchers fail to reject H_0, thinking the drug doesn't work, but it actually does, they've made a Type II error.

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

Answer :

Difference between Z-test and T-test :

- In Z-test, the population of standard deviation is known while In T-test , population of standard deviation is unknown.
- In Z-test , sample size is large (n >=30) while In T-test, sample size is small (n<=30).
- In both test, the data is approximately normally distributed.

When to use both test :

- Use a Z-test when dealing with large samples and known population variance.

- Use a T-test when the sample is small and/or the population variance is unknown.

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 :

```
# Question 6: Generate a binomial distribution and plot histogram
```

```
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 binomially distributed random numbers
data = np.random.binomial(n, p, size)

# Plot histogram
plt.hist(data, bins=range(n+2), edgecolor='black', align='left')
plt.title('Binomial Distribution Histogram (n=10, p=0.5)')
plt.xlabel('Number of Successes')
plt.ylabel('Frequency')
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

Output :

- The x-axis represents the **number of successes** (from 0 to 10).
- The y-axis shows how often each outcome occurred across 1000 samples.
- With p=0.5p = 0.5p=0.5, the histogram will look roughly **symmetric** and **centered around 5**, which is the expected mean of the binomial distribution.

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

# 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 = 50

# Calculate sample statistics
sample_mean = np.mean(sample_data)
sample_std = np.std(sample_data, ddof=0)  # Population std approximation
n = len(sample_data)

# Compute Z-statistic
z = (sample_mean - mu) / (sample_std / np.sqrt(n))

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

# Print results
print(f"Sample Mean = {sample_mean:.4f}")
print(f"Sample Std Dev = {sample_std:.4f}")
print(f"Z-statistic = {z:.4f}")
print(f"P-value = {p_value:.4f}")

# Interpretation
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: The sample mean is significantly different from 50.")
else:
    print("Fail to reject the null hypothesis: The sample mean is NOT significantly different from 50.")
```

<u>Output</u> :

```
Sample Mean = 50.0667
Sample Std Dev = 0.4590
Z-statistic = 0.8872
P-value = 0.3748
Fail to reject the null hypothesis: The sample mean is NOT significantly different 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. (Include your Python code and output in the code box below.)

Answer :

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

```
# Step 1: Simulate data from a normal distribution
np.random.seed(0)  # for reproducibility
mean = 100
std_dev = 15
n = 100

# Generate normal data
data = np.random.normal(loc=mean, scale=std_dev, size=n)

# Step 2: Calculate 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

# Calculate margin of error
z_critical = stats.norm.ppf(1 - alpha/2)  # Z for 95% confidence
margin_error = z_critical * (sample_std / np.sqrt(n))

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

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

# Step 3: Plot the data
plt.hist(data, bins=15, edgecolor='black', alpha=0.7)
plt.axvline(sample_mean, color='red', linestyle='--', label=f"Mean =
{sample_mean:.2f}")
plt.axvline(ci_lower, color='green', linestyle='--', label=f"Lower CI = {ci_lower:.2f}")
plt.axvline(ci_upper, color='green', linestyle='--', label=f"Upper CI = {ci_upper:.2f}")
plt.title("Histogram of Simulated Normal Data")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Output :

```
Sample Mean: 103.06
95% Confidence Interval: (100.98, 105.14)
```

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 :

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

# Function to calculate Z-scores
def calculate_z_scores(data):
    mean = np.mean(data)
    std = np.std(data, ddof=1)  # sample standard deviation
    z_scores = [(x - mean) / std for x in data]
    return z_scores

# Sample dataset
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]

# Calculate Z-scores
z_scores = calculate_z_scores(data)

# Output first few Z-scores for reference
print("First 5 Z-scores:")
for i in range(5):
    print(f"Data: {data[i]:.2f}, Z-score: {z_scores[i]:.2f}")

# Plot histogram of Z-scores
plt.hist(z_scores, bins=10, edgecolor='black', alpha=0.7)
plt.title("Histogram of Z-scores (Standardized Data)")
plt.xlabel("Z-score")
plt.ylabel("Frequency")
plt.grid(True, linestyle='--', alpha=0.6)
plt.axvline(0, color='red', linestyle='--', label='Mean (Z = 0)')
plt.legend()
plt.show()
```

Output :

```
First 5 Z-scores:
Data: 49.10, Z-score: -2.13
Data: 50.20, Z-score: 0.27
Data: 51.00, Z-score: 2.09
Data: 48.70, Z-score: -3.06
Data: 50.50, Z-score: 0.97
```