Q1. What are Type I and Type II errors in hypothesis testing, and how do they impact decision-making?
In hypothesis testing, **Type I and Type II errors** describe the two main ways a statistical decision can be incorrect. A **Type I error** occurs when a true null hypothesis is rejected, meaning the test indicates an effect or difference when none actually exists; this is often called a *false positive* and is controlled by the significance level (α). A **Type II error** happens when a false null hypothesis is not rejected, meaning the test fails to detect a real effect; this is known as a *false negative* and is represented by β, with statistical power equal to $1 - \beta$.

Q2What is the P-value in hypothesis testing, and how should it be interpreted in the context of the null hypothesis?
The **p-value** in hypothesis testing is the **probability of observing results as extreme as, or more extreme than, those obtained from the sample, assuming that the null hypothesis is true**. It measures how compatible the data are with the null hypothesis, not the probability that the null hypothesis itself is true. A **small p-value** indicates that such results would be unlikely if the null hypothesis were correct, providing evidence **against** the null hypothesis, while a **large p-value** suggests that the observed data are reasonably consistent with it.

Q3 :Explain the difference between a Z-test and a T-test, including when to use
A **Z-test** is used when the **population standard deviation is known** and the **sample size is large(greater than 30)**, while a **T-test** is used when the **population standard deviation is unknown**, especially for **small samples(less than 30)**. The T-test accounts for extra uncertainty using the t-distribution and is more commonly used in practice.

Q4 What is a confidence interval, and how does the margin of error influence its width and interpretation?
A **confidence interval** is a range of values calculated from sample data that is likely to contain the true population parameter (such as

a mean or proportion) with a specified level of confidence (for example, 95%). The **margin of error** determines how wide this interval is: a larger margin of error produces a wider interval, while a smaller margin of error results in a narrower one.

Q5Describe the purpose and assumptions of an ANOVA test. How does it extend hypothesis testing to more than two groups?

ANOVA tests whether the means of three or more groups are equal by comparing between-group and within-group variation, while controlling Type I error.

Q6 Write a Python program to perform a one-sample Z-test and interpret the result for a given dataset.

```
import numpy as np

from scipy.stats import norm


# Given data

data = [52, 55, 48, 50, 54, 51, 49]

mu_0 = 50        # population mean

sigma = 10      # population standard deviation

alpha = 0.05    # significance level


# Sample statistics

sample_mean = np.mean(data)

n = len(data)
```

```python
# Z-test statistic
z = (sample_mean - mu_0) / (sigma / np.sqrt(n))


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


# Results
print("Sample Mean:", sample_mean)
print("Z-value:", z)
print("P-value:", p_value)


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

Question 7:Simulate a dataset from a binomial distribution (n = 10, p = 0.5) using NumPy and plot the histogram.

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

Simulate data

data = np.random.binomial(n=10, p=0.5, size=1000)

Plot histogram

plt.hist(data)

plt.xlabel("Number of Successes")

plt.ylabel("Frequency")

plt.title("Histogram of Binomial Distribution (n = 10, p = 0.5)")

plt.show()

Question 8: Generate multiple samples from a non-normal distribution and implement the Central Limit Theorem using Python.

```
import numpy as np

import matplotlib.pyplot as plt


# Parameters

num_samples = 1000      # number of samples

sample_size = 50        # size of each sample


# Generate non-normal data (exponential distribution)

data = np.random.exponential(scale=1.0, size=(num_samples,
sample_size))
```

```
# Calculate sample means
sample_means = data.mean(axis=1)


# Plot histogram of sample means
plt.hist(sample_means)
plt.xlabel("Sample Mean")
plt.ylabel("Frequency")
plt.title("Central Limit Theorem using Exponential Distribution")
plt.show()
```

Question 9: Write a Python function to calculate and visualize the confidence interval for a sample mean

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


def confidence_interval_mean(data, confidence=0.95):
    data = np.array(data)
    mean = np.mean(data)
    std = np.std(data, ddof=1)
    n = len(data)
```

```python
    z = norm.ppf(1 - (1 - confidence) / 2)
    margin_error = z * (std / np.sqrt(n))

    lower = mean - margin_error
    upper = mean + margin_error

    # Plot confidence interval
    plt.errorbar(mean, 0, xerr=margin_error, fmt='o')
    plt.axvline(lower, linestyle='--')
    plt.axvline(upper, linestyle='--')
    plt.yticks([])
    plt.xlabel("Value")
    plt.title(f"{int(confidence*100)}% Confidence Interval for Mean")
    plt.show()

    return mean, lower, upper

# Example usage
sample_data = [12, 15, 14, 10, 13, 16, 11, 14]
confidence_interval_mean(sample_data)
```

Question 10: Perform a Chi-square goodness-of-fit test using Python to compare observed and expected distributions, and explain the outcome

```
import numpy as np

from scipy.stats import chisquare


# Observed and expected frequencies

observed = np.array([50, 30, 20])

expected = np.array([40, 40, 20])


# Chi-square goodness-of-fit test

chi_stat, p_value = chisquare(f_obs=observed, f_exp=expected)


print("Chi-square statistic:", chi_stat)

print("P-value:", p_value)
```