Double-click (or enter) to edit

Q13. Write a Python program to calculate the expected frequencies of a Chi-square test based on observed data

```
# prompt: Write a Python program to calculate the expected frequencies of a Chi-square test based on observed data

import numpy as np
from scipy.stats import chi2_contingency

def calculate_expected_frequencies(observed_data):
    """
    Calculates the expected frequencies for a Chi-square test of independence.

    Args:
        observed_data: A 2D list or numpy array representing the observed frequencies.

    Returns:
        A numpy array representing the expected frequencies.
    """
    chi2, p, dof, expected = chi2_contingency(observed_data)
    return expected

# Example usage
observed_data = [
    [20, 15],
    [10, 25]
]

expected_frequencies = calculate_expected_frequencies(observed_data)
print("Expected Frequencies:")
expected_frequencies
```

```
Expected Frequencies:
array([[15., 20.],
       [15., 20.]])
```

Q14. Perform a goodness-of-fit using Python to compare the observed data to an expected distribution

```
# prompt: Perform a goodness-of-fit using Python to compare the observed data to an expected distribution

from scipy.stats import chisquare

def goodness_of_fit(observed, expected):
    """
    Performs a Chi-square goodness-of-fit test.

    Args:
        observed: A list or numpy array of observed frequencies.
        expected: A list or numpy array of expected frequencies.

    Returns:
        A dictionary containing the chi-square statistic, p-value, and degrees of freedom.
    """

    chi2, p = chisquare(observed, f_exp=expected)
    dof = len(observed) - 1  # Degrees of freedom

    return {"chi2_statistic": chi2, "p_value": p, "degrees_of_freedom": dof}


# Example usage:
observed_frequencies = [15, 20, 25, 20, 20] #Example observed data
expected_frequencies = [20, 20, 20, 20, 20] #Example expected data (uniform distribution)

results = goodness_of_fit(observed_frequencies, expected_frequencies)
results
```

```
{'chi2_statistic': np.float64(2.5),
 'p_value': np.float64(0.6446357929354278),
 'degrees_of_freedom': 4}
```