Atharva Yadav
T23 - 101
DSPL ASSG 4 & 6

Question 1: In the population, the average IQ is 100 with a standard deviation of 15. A team of scientists want to test a new medication to see if it has either a positive or negative effect on intelligence, or not effect at all. A sample of 30 participants who have taken the medication has a mean of 140. Did the medication affect intelligence?

In [2]:
```
# Import necessary libraries
import numpy as np
import scipy.stats as stats

# Define the data
population_mean = 100
population_std = 15
sample_mean = 140
sample_size = 30
alpha = 0.05  # significance level

# Calculate the standard error of the mean
standard_error = population_std / np.sqrt(sample_size)

# Calculate the z-score
z_score = (sample_mean - population_mean) / standard_error

# Find the critical value for a two-tailed test at alpha = 0.05
critical_value = stats.norm.ppf(1 - alpha/2)

# Output results
print(f"Z-score: {z_score}")
print(f"Critical value: {critical_value}")

# Decision
if abs(z_score) > critical_value:
    print("Reject the null hypothesis: The medication had an effect on intelligence
else:
    print("Fail to reject the null hypothesis: The medication did not have an effec
```

```
Z-score: 14.60593486680443
Critical value: 1.959963984540054
Reject the null hypothesis: The medication had an effect on intelligence.
```

Question 2 A professor wants to know if her introductory statistics class has a good grasp of basic math. Six students are chosen at random form the call an given a math proficiency test. The professor wants the class to be able to score above 70 on the test. The six students get the following scores: 62, 92, 75, 68, 83, 95. Can the professor have 90% confidence that the mean score for the class on the test would be above 70.

In [3]:
```
import numpy as np
import scipy.stats as stats

# Data (scores from the test)
scores = np.array([62, 92, 75, 68, 83, 95])

# Sample size (n)
```

```
n = len(scores)

# Population mean (70, as per professor's desired threshold)
mu_0 = 70

# Calculate the sample mean and sample standard deviation
sample_mean = np.mean(scores)
sample_std = np.std(scores, ddof=1)  # Sample standard deviation

# Calculate the t-statistic
standard_error = sample_std / np.sqrt(n)
t_stat = (sample_mean - mu_0) / standard_error

# Degrees of freedom
df = n - 1

# Find the critical t-value for a one-tailed test at alpha = 0.10
alpha = 0.10
t_critical = stats.t.ppf(1 - alpha, df)

# Output results
print(f"Sample mean: {sample_mean:.2f}")
print(f"Sample standard deviation: {sample_std:.2f}")
print(f"Calculated t-statistic: {t_stat:.2f}")
print(f"Critical t-value (one-tailed, alpha=0.10): {t_critical:.2f}")

# Make the decision
if t_stat > t_critical:
    print("Reject the null hypothesis: The class's mean score is significantly abov
else:
    print("Fail to reject the null hypothesis: We cannot be 90% confident that the
```

```
Sample mean: 79.17
Sample standard deviation: 13.17
Calculated t-statistic: 1.71
Critical t-value (one-tailed, alpha=0.10): 1.48
Reject the null hypothesis: The class's mean score is significantly above 70.
```

Question 3 A clinic provides a program to help their clients lose weight and asks a consumer agency to investigate the effectiveness of the program. The agency takes a sample of 15 people, weighing each person in the sample before the program begins and 3 months later. The results a tabulated below. Determine is the program is effective.

In [4]:
```
import numpy as np
import scipy.stats as stats

# Data: Before and After weights
before = np.array([210, 205, 193, 182, 239, 164, 177, 222, 211, 187, 175, 181, 215,
after = np.array([197, 195, 191, 174, 226, 157, 171, 207, 202, 181, 164, 176, 211,

# Compute the differences
differences = before - after

# Perform a paired t-test
t_stat, p_value = stats.ttest_rel(before, after)
```

```python
# Compute mean and standard deviation of the differences
mean_diff = np.mean(differences)
std_diff = np.std(differences, ddof=1)  # Sample standard deviation

# Print results
print(f"Mean weight difference: {mean_diff:.2f} lbs")
print(f"Standard deviation of differences: {std_diff:.2f} lbs")
print(f"t-statistic: {t_stat:.2f}")
print(f"p-value: {p_value:.6f}")

# Conclusion
alpha = 0.05
if p_value < alpha:
    print("The weight-loss program is statistically effective (reject H0).")
else:
    print("The weight-loss program is not statistically effective (fail to reject H
```

```
Mean weight difference: 8.86 lbs
Standard deviation of differences: 4.13 lbs
t-statistic: 8.02
p-value: 0.000002
The weight-loss program is statistically effective (reject H0).
```

In [5]:

```python
import numpy as np
import pandas as pd

# Creating the dataset
data = {
    "Model": ["Model A", "Model B", "Model C"],
    "Trial 1": [3.5, 3.9, 3.5],
    "Trial 2": [3.4, 3.8, 3.3],
    "Trial 3": [3.8, 3.7, 3.6],
    "Trial 4": [3.5, 3.9, 3.5],
    "Trial 5": [3.4, 3.6, 3.8]
}

# Creating a DataFrame
df = pd.DataFrame(data)

# Calculating mean and standard deviation for each model
df["Mean Prediction"] = df.iloc[:, 1:].mean(axis=1)
df["Std Deviation"] = df.iloc[:, 1:].std(axis=1)

# Displaying the results
print(df)
```

```
     Model  Trial 1  Trial 2  Trial 3  Trial 4  Trial 5  Mean Prediction  \
0  Model A      3.5      3.4      3.8      3.5      3.4             3.52   
1  Model B      3.9      3.8      3.7      3.9      3.6             3.78   
2  Model C      3.5      3.3      3.6      3.5      3.8             3.54   

   Std Deviation  
0       0.146969  
1       0.116619  
2       0.162481  
```