Q1. Write a Python function that takes in two arrays of data and calculates the F-value for a variance ratio test. The function should return the F-value and the corresponding p-value for the test.

In [2]:
```
import scipy.stats as stats

def calculate_f_value(data1, data2):
    f_value = np.var(data1, ddof=1) / np.var(data2, ddof=1)
    df1 = len(data1) - 1
    df2 = len(data2) - 1
    p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, d
    return f_value, p_value
```

Q2. Given a significance level of 0.05 and the degrees of freedom for the numerator and denominator of an F-distribution, write a Python function that returns the critical F-value for a two-tailed test.

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

def calculate_critical_f(alpha, df1, df2):
    critical_f = stats.f.ppf(1 - alpha/2, df1, df2)
    return critical_f
```

Q3. Write a Python program that generates random samples from two normal distributions with known

variances and uses an F-test to determine if the variances are equal. The program should output the F- value, degrees of freedom, and p-value for the test.

In [5]:
```
import numpy as np
import scipy.stats as stats

# Generate random samples with known variances
np.random.seed(0)
sample1 = np.random.normal(0, np.sqrt(10), 20)
sample2 = np.random.normal(0, np.sqrt(15), 25)

# Perform F-test
f_value = np.var(sample1, ddof=1) / np.var(sample2, ddof=1)
df1 = len(sample1) - 1
df2 = len(sample2) - 1
p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, df1,

print("F-value:", f_value)
print("Degrees of Freedom:", df1, df2)
print("p-value:", p_value)
```

```
F-value: 0.318113689796604
Degrees of Freedom: 19 24
p-value: 0.013373067391727606
```

In [ ]:
```
Q4.The variances of two populations are known to be 10 and 15. A sample of 12 ob
each population. Conduct an F-test at the 5% significance level to determine if
significantly different.
```

In [6]:
```
import scipy.stats as stats

# Known variances
variance1 = 10
variance2 = 15
sample_size1 = 12
sample_size2 = 12

# Perform F-test
f_value = variance1 / variance2
df1 = sample_size1 - 1
df2 = sample_size2 - 1
p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, df1,

alpha = 0.05

if p_value < alpha:
    print("Reject the null hypothesis: Variances are significantly different.")
else:
    print("Fail to reject the null hypothesis: Variances are not significantly d
```

```
Fail to reject the null hypothesis: Variances are not significantly different.
```

Q5. A manufacturer claims that the variance of the diameter of a certain product is 0.005. A sample of 25 products is taken, and the sample variance is found to be 0.006. Conduct an F-test at the 1% significance level to determine if the claim is justified.

In [7]:
```
import scipy.stats as stats

# Claimed variance
claimed_variance = 0.005
sample_variance = 0.006
sample_size = 25

# Perform F-test
f_value = sample_variance / claimed_variance
df1 = sample_size - 1
df2 = 1  # Degrees of freedom for the claimed variance
p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, df1,

alpha = 0.01

if p_value < alpha:
    print("Reject the null hypothesis: The manufacturer's claim is not justified
else:
    print("Fail to reject the null hypothesis: The manufacturer's claim is justi
```

```
Fail to reject the null hypothesis: The manufacturer's claim is justified.
```

Q6. Write a Python function that takes in the degrees of freedom for the numerator and denominator of an F-distribution and calculates the mean and variance of the distribution. The function should return the mean and variance as a tuple.

In [9]:
```
import scipy.stats as stats

def calculate_f_distribution_mean_and_variance(df1, df2):
    mean = df2 / (df2 - 2)
```

```
    variance = (2 * (df2**2) * (df1 + df2 - 2)) / (df1 * (df2 - 2)**2 * (df2 - 4
    return mean, variance
```

Q7. A random sample of 10 measurements is taken from a normal population with unknown variance. The sample variance is found to be 25. Another random sample of 15 measurements is taken from another normal population with unknown variance, and the sample variance is found to be 20. Conduct an F-test at the 10% significance level to determine if the variances are significantly different.

In [10]:
```
import scipy.stats as stats

# Sample variances and sizes
variance1 = 25
sample_size1 = 10
variance2 = 20
sample_size2 = 15

# Perform F-test
f_value = variance1 / variance2
df1 = sample_size1 - 1
df2 = sample_size2 - 1
p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, df1,

alpha = 0.10  # 10% significance level

if p_value < alpha:
    print("Reject the null hypothesis: Variances are significantly different.")
else:
    print("Fail to reject the null hypothesis: Variances are not significantly d
```

```
Fail to reject the null hypothesis: Variances are not significantly different.
```

Q8. The following data represent the waiting times in minutes at two different restaurants on a Saturday night: Restaurant A: 24, 25, 28, 23, 22, 20, 27; Restaurant B: 31, 33, 35, 30, 32, 36. Conduct an F-test at the 5% significance level to determine if the variances are significantly different.

In [11]:
```
import scipy.stats as stats

# Waiting times at Restaurant A and Restaurant B
restaurant_A = [24, 25, 28, 23, 22, 20, 27]
restaurant_B = [31, 33, 35, 30, 32, 36]

# Perform F-test
variance_A = np.var(restaurant_A, ddof=1)
sample_size_A = len(restaurant_A)
variance_B = np.var(restaurant_B, ddof=1)
sample_size_B = len(restaurant_B)

f_value = variance_A / variance_B
df1 = sample_size_A - 1
df2 = sample_size_B - 1
p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, df1,

alpha = 0.05  # 5% significance level

if p_value < alpha:
```

```
    print("Reject the null hypothesis: Variances of waiting times are significan
else:
    print("Fail to reject the null hypothesis: Variances of waiting times are no
```

```
Fail to reject the null hypothesis: Variances of waiting times are not signific
antly different.
```

Q9. The following data represent the test scores of two groups of students: Group A: 80, 85, 90, 92, 87, 83; Group B: 75, 78, 82, 79, 81, 84. Conduct an F-test at the 1% significance level to determine if the variances are significantly different.

In [12]:
```
import scipy.stats as stats

# Test scores of Group A and Group B
group_A = [80, 85, 90, 92, 87, 83]
group_B = [75, 78, 82, 79, 81, 84]

# Perform F-test
variance_A = np.var(group_A, ddof=1)
sample_size_A = len(group_A)
variance_B = np.var(group_B, ddof=1)
sample_size_B = len(group_B)

f_value = variance_A / variance_B
df1 = sample_size_A - 1
df2 = sample_size_B - 1
p_value = 2 * min(stats.f.cdf(f_value, df1, df2), 1 - stats.f.cdf(f_value, df1,

alpha = 0.01  # 1% significance level

if p_value < alpha:
    print("Reject the null hypothesis: Variances of test scores are significantl
else:
    print("Fail to reject the null hypothesis: Variances of test scores are not
```

```
Fail to reject the null hypothesis: Variances of test scores are not significan
tly different.
```