Here are Python solutions for each of the questions in the assignment:

**Q1: Python function to calculate the F-value and p-value for a variance ratio test.**

import numpy as np

from scipy import stats

def f\_test(array1, array2):

var1 = np.var(array1, ddof=1)

var2 = np.var(array2, ddof=1)

# F-value calculation

f\_value = var1 / var2 if var1 > var2 else var2 / var1

# Degrees of freedom

df1 = len(array1) - 1

df2 = len(array2) - 1

# p-value calculation

p\_value = stats.f.sf(f\_value, df1, df2)

return f\_value, p\_value

**Q2: Python function to return the critical F-value for a two-tailed test.**

def critical\_f\_value(df1, df2, alpha=0.05):

f\_critical = stats.f.ppf(1 - alpha/2, df1, df2) # Two-tailed test

return f\_critical

**Q3: Python program to generate random samples from two normal distributions and conduct an F-test.**

def f\_test\_random\_samples(mean1, var1, mean2, var2, size1, size2):

# Generate random samples

sample1 = np.random.normal(mean1, np.sqrt(var1), size1)

sample2 = np.random.normal(mean2, np.sqrt(var2), size2)

# Perform F-test

f\_value, p\_value = f\_test(sample1, sample2)

df1 = len(sample1) - 1

df2 = len(sample2) - 1

return f\_value, df1, df2, p\_value

**Q4: F-test to determine if the variances are significantly different.**

def f\_test\_known\_variances(var1, var2, n1, n2, alpha=0.05):

f\_value = var1 / var2 if var1 > var2 else var2 / var1

df1 = n1 - 1

df2 = n2 - 1

# Critical F-value for two-tailed test

f\_critical = critical\_f\_value(df1, df2, alpha)

# Perform F-test

p\_value = stats.f.sf(f\_value, df1, df2)

return f\_value, df1, df2, f\_critical, p\_value

**Q5: Conduct an F-test to determine if the claim about the variance is justified.**

def f\_test\_claimed\_variance(claimed\_var, sample\_var, sample\_size, alpha=0.01):

f\_value = sample\_var / claimed\_var

df1 = sample\_size - 1

df2 = df1 # For a single sample, the degrees of freedom are equal.

# Critical F-value

f\_critical = critical\_f\_value(df1, df2, alpha)

# Perform F-test

p\_value = stats.f.sf(f\_value, df1, df2)

return f\_value, df1, df2, f\_critical, p\_value

**Q6: Python function to calculate the mean and variance of an F-distribution.**

def f\_distribution\_mean\_variance(df1, df2):

mean = df2 / (df2 - 2) if df2 > 2 else None

variance = (2 \* df2\*\*2 \* (df1 + df2 - 2)) / (df1 \* (df2 - 2)\*\*2 \* (df2 - 4)) if df2 > 4 else None

return mean, variance

**Q7: Conduct an F-test to determine if the variances are significantly different for two populations with unknown variances.**

def f\_test\_two\_unknown\_variances(var1, var2, n1, n2, alpha=0.10):

f\_value = var1 / var2 if var1 > var2 else var2 / var1

df1 = n1 - 1

df2 = n2 - 1

# Critical F-value

f\_critical = critical\_f\_value(df1, df2, alpha)

# Perform F-test

p\_value = stats.f.sf(f\_value, df1, df2)

return f\_value, df1, df2, f\_critical, p\_value

**Q8: Conduct an F-test for variances at the 5% significance level (for Restaurant A and Restaurant B data).**

def f\_test\_restaurant\_data(data\_a, data\_b, alpha=0.05):

f\_value, p\_value = f\_test(data\_a, data\_b)

df1 = len(data\_a) - 1

df2 = len(data\_b) - 1

f\_critical = critical\_f\_value(df1, df2, alpha)

return f\_value, df1, df2, f\_critical, p\_value

**Q9: Conduct an F-test for variances at the 1% significance level (for Group A and Group B test scores).**

def f\_test\_group\_scores(group\_a, group\_b, alpha=0.01):

f\_value, p\_value = f\_test(group\_a, group\_b)

df1 = len(group\_a) - 1

df2 = len(group\_b) - 1

f\_critical = critical\_f\_value(df1, df2, alpha)

return f\_value, df1, df2, f\_critical, p\_value

These functions allow you to compute F-values, p-values, critical F-values, and make inferences using the F-test on various datasets and hypotheses.