assignment=35

July 29, 2023

1 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.

[1]:

```
import numpy as np
from scipy.stats import f

def variance_ratio_test(x, y):
    """
    Calculate the F-value and p-value for a variance ratio test.

    Args:
    x (array-like): First array of data
    y (array-like): Second array of data

    Returns:
    F (float): The F-value for the variance ratio test
    p_value (float): The p-value for the variance ratio test
    """
    n_x = len(x)
    n_y = len(y)
    var_x = np.var(x, ddof=1)
    var_y = np.var(y, ddof=1)
    F = var_x / var_y
    p_value = f.sf(F, n_x-1, n_y-1)
    return F, p_value

data1 = np.random.randint(100, size=(50))
data2 = np.random.randint(100, size=(50))

f_val, p_val= variance_ratio_test(data1, data2)
print("The f_value and the p_value are",f_val,"and",p_val,"respectively.")
```

```
The f_value and the p_value are 1.1243855294428309 and 0.3415912559806815
respectively.
```

The function first calculates the sample variances for the two input arrays x and y using the np.var function with the ddof parameter set to 1, which specifies that the denominator of the variance calculation should be n-1 instead of n to account for the fact that we are estimating the population variance from a sample. The function then calculates the F-value by dividing the sample variance of x by the sample variance of y. Then, the function calculates the p-value for the F-value using the f.sf function from the scipy.stats module. The f.sf function returns the survival function, which gives the probability that a random variable with an F-distribution with the specified degrees of freedom is greater than or equal to the given F-value. To obtain the p-value, we need to subtract this probability from 1.

2 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.

[2]:

```
from scipy.stats import f


def critical_f_value(alpha, dfn, dfd):
    """
    Returns the critical F-value for a two-tailed test given a significance␣
 ↪level and
    the degrees of freedom for the numerator and denominator of an␣
 ↪F-distribution.

    Args:
    alpha (float): The significance level
    dfn (int): The degrees of freedom for the numerator
    dfd (int): The degrees of freedom for the denominator

    Returns:
    crit_f (float): The critical F-value
    """
    crit_f = f.ppf(alpha/2, dfn, dfd)
    return crit_f

crit_f_value = critical_f_value(0.05, 5, 4)
print("The critical F-value for a two-tailed test is", crit_f_value)
```

```
The critical F-value for a two-tailed test is 0.13535672229749918
```

The function uses the f.ppf function from the scipy.stats module to calculate the critical F-value for a two-tailed test. The f.ppf function returns the percent point function (inverse of the cumulative distribution function) of the F-distribution for a given probability (alpha/2 in this case) and degrees of freedom for the numerator and denominator of the F-distribution. The returned value is the F-value such that the probability of observing a larger F-value is equal to alpha/2.

3 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.

[3]:

```python
import numpy as np
from scipy.stats import f

# Set the random seed for reproducibility
np.random.seed(123)

# Generate random samples from two normal distributions with known variances
n1 = 30
n2 = 25
mean1 = 5
mean2 = 5
var1 = 4
var2 = 9
x = np.random.normal(mean1, np.sqrt(var1), n1)
y = np.random.normal(mean2, np.sqrt(var2), n2)

# Calculate the F-value and p-value for the variance ratio test
dfn = n1 - 1
dfd = n2 - 1
F = np.var(x, ddof=1) / np.var(y, ddof=1)
p_value = f.sf(F, dfn, dfd)

# Print the results
print("F-value: {:.4f}".format(F))
print("Degrees of freedom: ({}, {})".format(dfn, dfd))
print("p-value: {:.4f}".format(p_value))
```

```
F-value: 0.4164
Degrees of freedom: (29, 24)
p-value: 0.9872
```

The program first sets the random seed for reproducibility using the np.random.seed function. Then, it generates random samples from two normal distributions with known variances using the np.random.normal function. The parameters for the distributions are specified as follows: mean1 and var1 for the first distribution, and mean2 and var2 for the second distribution. The sample sizes for the two distributions are specified as n1 and n2, respectively. The program then calculates the F-value and p-value for the variance ratio test using the np.var and f.sf functions from the numpy and scipy.stats modules, respectively. The degrees of freedom for the numerator and denominator of the F-distribution are calculated as dfn = n1 - 1 and dfd = n2 - 1, respectively. The program prints the results of the test, including the F-value, degrees of freedom, and p-value, using the print function with formatted strings.

4 Q4.The variances of two populations are known to be 10 and 15. A sample of 12 observations is taken from each population. Conduct an F-test at the 5% significance level to determine if the variances are significantly different.

[4]:

```
import numpy as np
from scipy.stats import f

# Set the significance level
alpha = 0.05

# Set the known variances
var1 = 10
var2 = 15

# Set the sample sizes
n1 = 12
n2 = 12

# Generate random samples from two normal distributions with known variances
mean1 = 0
mean2 = 0
x = np.random.normal(mean1, np.sqrt(var1), n1)
y = np.random.normal(mean2, np.sqrt(var2), n2)

# Calculate the F-value and p-value for the variance ratio test
dfn = n1 - 1
dfd = n2 - 1
F = np.var(x, ddof=1) / np.var(y, ddof=1)
p_value = f.sf(F, dfn, dfd)

# Determine if the variances are significantly different
if p_value < alpha:
    print("The variances are significantly different (p-value = {:.4f})".
 ↪format(p_value))
else:
    print("The variances are not significantly different (p-value = {:.4f})".
 ↪format(p_value))
```

```
The variances are not significantly different (p-value = 0.7823)
```

First, we set the significance level to 0.05 using the variable alpha. We set the known variances of the two populations to 10 and 15 using the variables var1 and var2. We also set the sample sizes to 12 using the variables n1 and n2. Then, we generate random samples from two normal distributions with known variances using the np.random.normal function. The means of the two distributions are set to 0, since the means are not relevant for the F-test. We then calculate the F-value and p-value for the variance ratio test using the np.var and f.sf functions from the numpy and scipy.stats modules, respectively. The degrees of freedom for the numerator and denominator of the F-distribution are calculated as dfn = n1 - 1 and dfd = n2 - 1, respectively. We determine if the variances are significantly different by comparing the p-value to the significance level. If the p-value is less than the significance level, we conclude that the variances are significantly different. Otherwise, we conclude that the variances are not significantly different.

5 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.

[5]:

```python
import numpy as np
from scipy.stats import f

# Set the significance level
alpha = 0.01

# Set the claimed variance
var_claimed = 0.005

# Set the sample size
n = 25

# Set the sample variance
var_sample = 0.006

# Calculate the F-value and p-value for the variance ratio test
dfn = n - 1
dfd = n - 1
F = var_sample / var_claimed
p_value = f.sf(F, dfn, dfd)

# Determine if the claim is justified
if p_value < alpha:
    print("The claim is not justified (p-value = {:.4f})".format(p_value))
else:
    print("The claim is justified (p-value = {:.4f})".format(p_value))
```

```
The claim is justified (p-value = 0.3294)
```

we set the significance level to 0.01 using the variable alpha. we set the claimed variance of the diameter of the product to 0.005 using the variable var_claimed. We also set the sample size to 25 using the variable n. we set the sample variance of the diameter of the product to 0.006 using the variable var_sample. We then calculate the F-value and p-value for the variance ratio test using the formula F = var_sample / var_claimed and the f.sf function from the scipy.stats module. The degrees of freedom for the numerator and denominator of the F-distribution are both equal to n -

1. We determine if the claim is justified by comparing the p-value to the significance level. If the p-value is less than the significance level, we conclude that the claim is not justified. Otherwise, we conclude that the claim is justified.

6 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.

[6]:

```python
def f_distribution_mean_var(dfn, dfd):
    """
    Calculate the mean and variance of an F-distribution given the degrees of␣
 ↪freedom for the numerator and denominator.

    Parameters:
    dfn (int): Degrees of freedom for the numerator.
    dfd (int): Degrees of freedom for the denominator.

    Returns:
    tuple: Mean and variance of the F-distribution.
    """
    mean = dfd / (dfd - 2)
    variance = (2 * dfd ** 2 * (dfn + dfd - 2)) / \
        (dfn * (dfd - 2) ** 2 * (dfd - 4))

    return mean, variance


mean, variance = f_distribution_mean_var(5, 7)

print("The mean and the variance of the F-distribution are ",
      mean, "and", variance, "respectively.")
```

```
The mean and the variance of the F-distribution are  1.4 and 2.6133333333333333
respectively.
```

The f_distribution_mean_var function takes in two parameters, dfn and dfd, which represent the degrees of freedom for the numerator and denominator of the F-distribution, respectively. The mean of the F-distribution is calculated as dfd / (dfd - 2). The variance of the F-distribution is calculated using the formula (2 * dfd ** 2 * (dfn + dfd - 2)) / (dfn * (dfd - 2) ** 2 * (dfd - 4)). The function returns a tuple of the mean and variance of the F-distribution.

7 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

[7]:

```python
import numpy as np
from scipy.stats import f

# Set the significance level
alpha = 0.1

# Set the sample sizes
n1 = 10
n2 = 15

# Set the sample variances
s1 = 25
s2 = 20

# Calculate the F-value and p-value for the variance ratio test
dfn = n1 - 1
dfd = n2 - 1
F = s1 / s2
p_value = f.sf(F, dfn, dfd)

# Determine if the variances are significantly different
if p_value < alpha:
    print("The variances are significantly different (p-value = {:.4f})".
  ↪format(p_value))
else:
    print("The variances are not significantly different (p-value = {:.4f})".
  ↪format(p_value))
```

```
The variances are not significantly different (p-value = 0.3416)
```

First, we set the significance level to 0.1 using the variable alpha. We set the sample sizes to 10 and 15 using the variables n1 and n2, respectively. We also set the sample variances to 25 and 20 using the variables s1 and s2, respectively. We then calculate the F-value and p-value for the variance ratio test using the formula F = s1 / s2 and the f.sf function from the scipy.stats module. The degrees of freedom for the numerator and denominator of the F-distribution are n1 - 1 and n2 - 1, respectively. We determine if the variances are significantly different by comparing the p-value to the significance level. If the p-value is less than the significance level, we conclude that the variances are significantly different. Otherwise, we conclude that the variances are not significantly different.

8 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.

[8]:

```
import numpy as np
from scipy.stats import f

# Set the significance level
alpha = 0.05

# Set the data
a = np.array([24, 25, 28, 23, 22, 20, 27])
b = np.array([31, 33, 35, 30, 32, 36])

# Calculate the sample variances
var_a = np.var(a, ddof=1)
var_b = np.var(b, ddof=1)

# Calculate the F-value and p-value for the variance ratio test
F = var_a / var_b
dfn = len(a) - 1
dfd = len(b) - 1
p_value = f.sf(F, dfn, dfd)

# Determine if the variances are significantly different
if p_value < alpha:
    print("The variances are significantly different (p-value = {:.4f})".
  ↪format(p_value))
else:
    print("The variances are not significantly different (p-value = {:.4f})".
  ↪format(p_value))
```

```
The variances are not significantly different (p-value = 0.3487)
```

First, we set the significance level to 0.05 using the variable alpha. We set the data for each restaurant using the variables a and b, respectively. We then calculate the sample variances using the np.var function with ddof=1 to account for the fact that we are estimating the population variance from a sample. We calculate the F-value and p-value for the variance ratio test using the formula F = var_a / var_b and the f.sf function from the scipy.stats module. The degrees of freedom for the numerator and denominator of the F-distribution are len(a) - 1 and len(b) - 1, respectively. We determine if the variances are significantly different by comparing the p-value to the significance level. If the p-value is less than the significance level, we conclude that the variances are significantly different. Otherwise, we conclude that the variances are not significantly different.

9 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.

[9]:

```
import numpy as np
from scipy.stats import f

# Set the significance level
alpha = 0.01

# Set the data
a = np.array([80, 85, 90, 92, 87, 83])
b = np.array([75, 78, 82, 79, 81, 84])

# Calculate the sample variances
var_a = np.var(a, ddof=1)
var_b = np.var(b, ddof=1)

# Calculate the F-value and p-value for the variance ratio test
F = var_a / var_b
dfn = len(a) - 1
dfd = len(b) - 1
p_value = f.sf(F, dfn, dfd)

# Determine if the variances are significantly different
if p_value < alpha:
    print("The variances are significantly different (p-value = {:.4f})".
 ↪format(p_value))
else:
    print("The variances are not significantly different (p-value = {:.4f})".
 ↪format(p_value))
```

```
The variances are not significantly different (p-value = 0.2416)
```

First, we set the significance level to 0.01 using the variable alpha. we set the data for each group using the variables a and b, respectively. We then calculate the sample variances using the np.var function with ddof=1 to account for the fact that we are estimating the population variance from a sample. We calculate the F-value and p-value for the variance ratio test using the formula F = var_a / var_b and the f.sf function from the scipy.stats module. The degrees of freedom for the numerator and denominator of the F-distribution are len(a) - 1 and len(b) - 1, respectively. We determine if the variances are significantly different by comparing the p-value to the significance level. If the p-value is less than the significance level, we conclude that the variances are significantly different. Otherwise, we conclude that the variances are not significantly different.

[ ]:

```
```