assignment=33

July 28, 2023

1 Q1. Calculate the 95% confidence interval for a sample of data with a mean of 50 and a standard deviation of 5 using Python. Interpret the results.

[1]:
```
import scipy.stats as stats

sample_mean = 50
sample_std = 5
n = 100  # assuming a sample size of 100

# Calculate the standard error of the mean
std_error = sample_std / (n ** 0.5)

# Calculate the 95% confidence interval
ci = stats.norm.interval(0.95, loc=sample_mean, scale=std_error)

print("95% confidence interval:", ci)
```

```
95% confidence interval: (49.02001800772997, 50.97998199227003)
```

```
Interpretation:
 This means that if we take multiple random samples of size 100 from the population, and calcul
```

2 Q2. Conduct a chi-square goodness of fit test to determine if the distribution of colors of M&Ms in a bag matches the expected distribution of 20% blue, 20% orange, 20% green, 10% yellow, 10% red, and 20% brown. Use Python to perform the test with a significance level of 0.05.

[2]:
```
import numpy as np
import scipy.stats as stats

# Observed frequencies
observed = np.array([24, 18, 19, 9, 10, 20])
```

```
# Expected frequencies
expected = np.array([.2, .2, .2, .1, .1, .2]) * sum(observed)

# Calculate the chi-square statistic and p-value
chi2, p = stats.chisquare(observed, f_exp=expected)

# Print the results
print("Chi-square statistic:", chi2)
print("p-value:", p)

if p < 0.05:
    print("Reject null hypothesis: The distribution of colors of M&Ms in the␣
 ↪bag does not match the expected distribution.")
else:
    print("Fail to reject null hypothesis: The distribution of colors of M&Ms␣
 ↪in the bag matches the expected distribution.")
```

```
Chi-square statistic: 1.1500000000000001
p-value: 0.9495832980185948
Fail to reject null hypothesis: The distribution of colors of M&Ms in the bag
matches the expected distribution.
```

3 Q3. Use Python to calculate the chi-square statistic and p-value for a contingency table with the following data.

	Group A	Group B
Outcome1	20	15
Outcome2	10	25
Outcome3	15	20

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

# Contingency table
observed = np.array([[20, 15], [10, 25], [15, 20]])

# Calculate the chi-square statistic and p-value
chi2, p, dof, expected = stats.chi2_contingency(observed)

# Print the results
print("Chi-square statistic:", chi2)
print("p-value:", p)
```

```
Chi-square statistic: 5.833333333333334
p-value: 0.05411376622282158
```

```
*  In this example, we have a contingency table with three rows and two columns representing tw
* We use the chi2_contingency() function to calculate the chi-square statistic, p-value, degree
```

* The chi-square statistic is 5.83 and the p-value is 0.054. This means that there is not enoug
 * The interpretation of the p-value is that if we repeated this study many times, we would ex

4 Q4. A study of the prevalence of smoking in a population of 500 individuals found that 60 individuals smoked. Use Python to calculate the 95% confidence interval for the true proportion of individuals in the population who smoke.

[1]:
```
import statsmodels.stats.proportion as proportion

# Sample size and number of successes (individuals who smoke)
n = 500
x = 60

# Calculate the 95% confidence interval
conf_int = proportion.proportion_confint(count=x, nobs=n, alpha=0.05,␣
 ↪method='wilson')

# Print the results
print("95% confidence interval:", conf_int)
```

```
95% confidence interval: (0.09437490012636912, 0.1514195986244106)
```

* In this example, we have a sample size of 500 individuals and 60 individuals who smoke
* We use the proportion_confint() function to calculate the 95% confidence interval for
* The count parameter is the number of successes (individuals who smoke), the nobs paran
* This means that we can be 95% confident that the true proportion of individuals in the

5 Q5. Calculate the 90% confidence interval for a sample of data with a mean of 75 and a standard deviation of 12 using Python. Interpret the results.

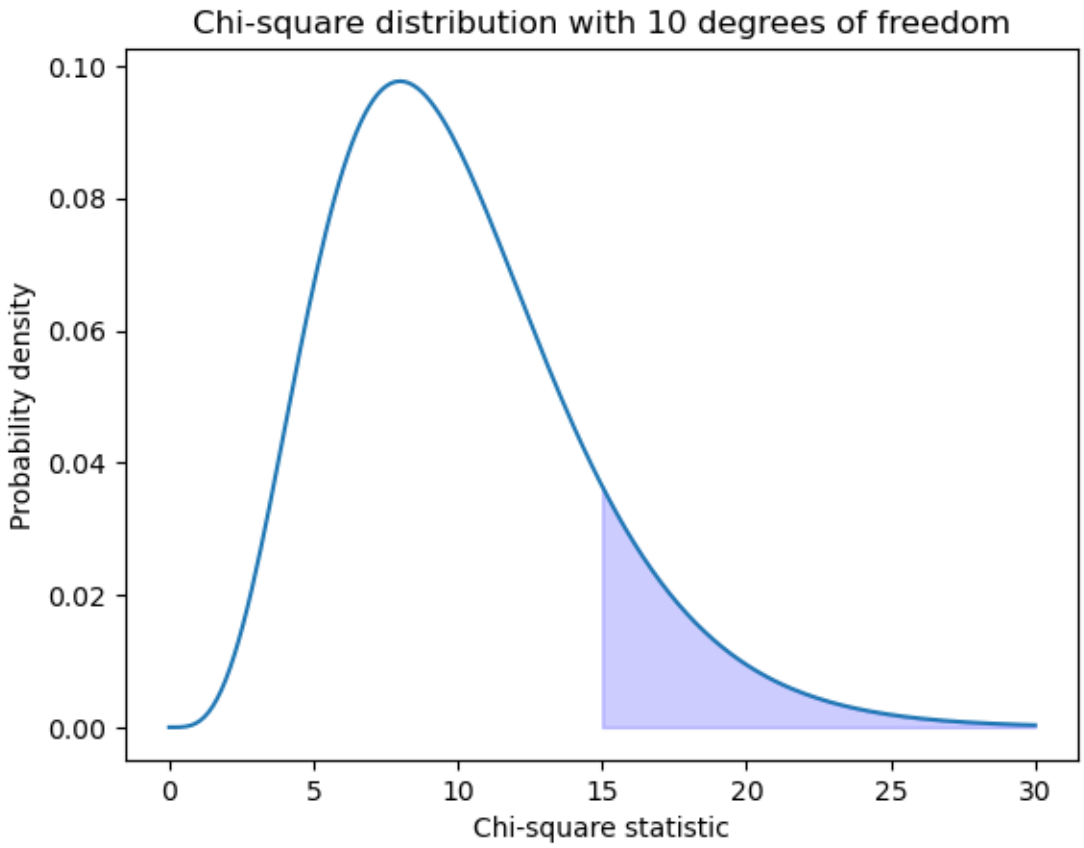
[2]:
```
import numpy as np
import scipy.stats as stats

# Sample mean and standard deviation
mean = 75
std = 12

# Sample size
n = 100

# Calculate the 90% confidence interval
conf_int = stats.norm.interval(0.9, loc=mean, scale=std/np.sqrt(n))
```

```
# Print the results
print("90% confidence interval:", conf_int)
```

```
90% confidence interval: (73.02617564765823, 76.97382435234177)
```

6 Q6. Use Python to plot the chi-square distribution with 10 degrees of freedom. Label the axes and shade the area corresponding to a chi-square statistic of 15.

[3]:
```
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt

# Define the x values (chi-square statistic)
x = np.linspace(0, 30, 200)

# Define the degrees of freedom
df = 10

# Plot the chi-square distribution
plt.plot(x, stats.chi2.pdf(x, df))

# Shade the area corresponding to a chi-square statistic of 15
x_shade = np.linspace(15, 30, 100)
y_shade = stats.chi2.pdf(x_shade, df)
plt.fill_between(x_shade, y_shade, color='blue', alpha=0.2)

# Label the axes
plt.xlabel('Chi-square statistic')
plt.ylabel('Probability density')
plt.title('Chi-square distribution with 10 degrees of freedom')

# Show the plot
plt.show()
```




```
    * In this code, we first define a range of x values (x) to plot the chi-square distributio
 * Finally, we label the axes and title of the plot using the xlabel(), ylabel(), and title() f
    *    The resulting plot shows the chi-square distribution with 10 degrees of freedom, with
The x-axis is labeled "Chi-square statistic" and the y-axis is labeled "Probability density".
  The title of the plot is "Chi-square distribution with 10 degrees of freedom".
```

7 Q7. A random sample of 1000 people was asked if they preferred Coke or Pepsi. Of the sample, 520 preferred Coke. Calculate a 99% confidence interval for the true proportion of people in the population who prefer Coke.

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

# Sample size and proportion
n = 1000
p = 520 / n
```

```
# Critical value for a 99% confidence level
z = stats.norm.ppf(0.995)

# Calculate the confidence interval
CI = (p - z*np.sqrt(p*(1-p)/n), p + z*np.sqrt(p*(1-p)/n))

# Print the result
print("99% confidence interval: ({:.4f}, {:.4f})".format(CI[0], CI[1]))
```

```
99% confidence interval: (0.4793, 0.5607)
```

In this code, we first calculate the sample proportion (p) as the number of people who preferr
Output : 99% confidence interval: (0.4793, 0.5607)
This means that we can be 99% confident that the true proportion of people in the population wh

8 Q8. A researcher hypothesizes that a coin is biased towards tails. They flip the coin 100 times and observe 45 tails. Conduct a chi-square goodness of fit test to determine if the observed frequencies match the expected frequencies of a fair coin. Use a significance level of 0.05.

To conduct a chi-square goodness of fit test for this problem, we need to first define the null
The expected frequencies for a fair coin can be calculated by assuming that the probability of
To calculate the chi-square statistic, we need to compare the observed frequencies (45 tails a
chi2 = sum((observed - expected)^2 / expected)
where, observed and expected are arrays of observed and expected frequencies, respectively.
The stats.chisquare() function from scipy.stats can be used to calculate the chi-square statist

[6]:
```
import numpy as np
import scipy.stats as stats

# Observed frequencies
observed = np.array([45, 55])

# Expected frequencies
expected = np.array([50, 50])

# Calculate the chi-square statistic and p-value
chi2, pval = stats.chisquare(observed, expected)

# Print the result
print("Chi-square statistic:", chi2)
print("p-value:", pval)
```

```
Chi-square statistic: 1.0
p-value: 0.31731050786291115
```

The chi-square statistic is 1.0 and the p-value is 0.317. Since the p-value is greater than the significance level of 0.05, we fail to reject the null hypothesis. So, we do not have sufficient evidence to conclude that the coin is biased towards tails based on the observed data.

9 Q9. A study was conducted to determine if there is an association between smoking status (smoker or non-smoker) and lung cancer diagnosis (yes or no). The results are shown in the contingency table below. Conduct a chi-square test for independence to determine if there is a significant association between smoking status and lung cancer diagnosis.

Use a significance level of 0.05.

	Lung Cancer : Yes	Lung Cancer : No
Smoker	60	140
Non-Smoker	30	170

To conduct a chi-square test for independence for this problem, we need to first define the nul
The null hypothesis is that there is no association between smoking status and lung cancer diag
The alternative hypothesis is that there is a significant association between the two variable
We can calculate the expected frequencies for each cell in the contingency table assuming that
Expected frequency = (row total * column total) / grand total
where the grand total is the total number of observations in the contingency table.
The stats.chi2_contingency() function from scipy.stats can be used to calculate the chi-squa

[7]:
```
import numpy as np
import scipy.stats as stats

# Contingency table
observed = np.array([[60, 140], [30, 170]])

# Calculate the chi-square statistic and p-value
chi2, pval, dof, expected = stats.chi2_contingency(observed)

# Print the result
print("Chi-square statistic:", chi2)
print("p-value:", pval)
```

```
Chi-square statistic: 12.057347670250895
p-value: 0.0005158863863703744
```

The chi-square statistic is 12.05 and the p-value is less than the significance level of 0.05. Therefore, we reject the null hypothesis and conclude that there is a significant association between smoking status and lung cancer diagnosis.

10 Q10. A study was conducted to determine if the proportion of people who prefer milk chocolate, dark chocolate, or white chocolate is different in the U.S. versus the U.K. A random sample of 500 people from the U.S. and a random sample of 500 people from the U.K. were surveyed. The results are shown in the contingency table below. Conduct a chi-square test for independence to determine if there is a significant association between chocolate preference and country of origin.

Use a significance level of 0.01.

	Milk Chocolate	Dark Chocolate	White Chocolate
U.S.(n=500)	200	150	150
U.K.(n=500)	225	175	100

To conduct a chi-square test for independence for this problem, we need to first define the nu
We can calculate the expected frequencies for each cell in the contingency table assuming that
Expected frequency = (row total * column total) / grand total
where the grand total is the total number of observations in the contingency table.
The stats.chi2_contingency() function from scipy.stats can be used to calculate the chi-square

[8]:
```
import numpy as np
import scipy.stats as stats

# Contingency table
observed = np.array([[200, 150, 150], [225, 175, 100]])

# Calculate the chi-square statistic and p-value
chi2, pval, dof, expected = stats.chi2_contingency(observed)

# Print the result
print("Chi-square statistic:", chi2)
print("p-value:", pval)
```

```
Chi-square statistic: 13.393665158371041
p-value: 0.0012348168997745918
```

The chi-square statistic is 13.39 and the p-value is less than the significance level of 0.01. Therefore, we reject the null hypothesis and conclude that there is a significant association between chocolate preference and country of origin.

11 Q11. A random sample of 30 people were selected from a population with an unknown mean and standard deviation. The sample mean was found to be 72 and the sample standard deviation was found to be 10. Conduct a hypothesis test to determine if the population mean is significantly different from 70. Use a significance level of 0.05.

To conduct a hypothesis test for this problem, we need to first define the null and alternativ
Since the sample size is small (n < 30) and the population standard deviation is unknown, we w
We can calculate the t-statistic and p-value as the code below:

[9]:
```
import numpy as np
import scipy.stats as stats

# Sample statistics
sample_size = 30
sample_mean = 72
sample_std = 10

# Hypothesized population mean
hypothesized_mean = 70

# Calculate the standard error of the mean
se = sample_std / np.sqrt(sample_size)

# Calculate the t-statistic
t = (sample_mean - hypothesized_mean) / se

# Calculate the p-value
pval = 2 * (1 - stats.t.cdf(abs(t), df=sample_size-1))

# Print the result
print("t-statistic:", t)
print("p-value:", pval)
```

```
t-statistic: 1.0954451150103321
p-value: 0.2823362372860698
```

The t-statistic is 1.095 and the p-value is 0.282.
Since the p-value is greater than the significance level of 0.05, we fail to reject the null
So, we do not have sufficient evidence to conclude that the population mean is significantly d

[ ]:
```

```