statistics-advance-5

August 13, 2023

[ ]:
```
# 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

# Given data
mean = 50
std_dev = 5
confidence_level = 0.95
sample_size = 100  # Adjust this according to your actual sample size

# Calculate the margin of error
z_score = stats.norm.ppf(1 - (1 - confidence_level) / 2)
margin_of_error = z_score * (std_dev / (sample_size ** 0.5))

# Calculate the confidence interval
lower_bound = mean - margin_of_error
upper_bound = mean + margin_of_error

# Print the results
print("95% Confidence Interval: [{:.2f}, {:.2f}]".format(lower_bound,␣
 ↪upper_bound))
```

```
95% Confidence Interval: [49.02, 50.98]
```

0.1 Interpretation:

A 95% confidence interval means that if you were to take multiple samples from the same population and calculate the confidence interval for each sample, approximately 95% of those intervals would contain the true population mean. In this case, the calculated confidence interval is [48.99, 51.01], which means that we are 95% confident that the true population mean lies within this range.

[ ]:
```
# 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 scipy.stats as stats

# Observed frequencies (actual counts) of each color
observed_freq = [35, 40, 15, 10, 8, 12]  # Replace these values with your␣
 ↪actual data

# Expected frequencies based on the expected distribution
expected_freq = [0.20, 0.20, 0.20, 0.10, 0.10, 0.20]
total_observed = sum(observed_freq)
expected_freq = [total_observed * p for p in expected_freq]

# Perform the chi-square goodness of fit test
chi2_statistic, p_value = stats.chisquare(f_obs=observed_freq,␣
 ↪f_exp=expected_freq)

# Significance level
alpha = 0.05

# Print the results
print("Chi-square statistic:", chi2_statistic)
print("p-value:", p_value)

if p_value < alpha:
    print("Reject the null hypothesis. The distributions are significantly␣
 ↪different.")
else:
    print("Fail to reject the null hypothesis. The distributions are not␣
 ↪significantly different.")
```

```
Chi-square statistic: 26.749999999999996
p-value: 6.380546931227065e-05
Reject the null hypothesis. The distributions are significantly different.
```

[ ]:

```
# Q3. Use Python to calculate the chi-square statistic and p-value for a␣
 ↪contingency table with the following
# Data:
#            Group A   Group B
#Outcome 1     20         15
#Outcome 2     10         25
#Outcome 3     15         20
```

[3]:

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

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

# Perform the chi-square test
chi2_statistic, p_value, dof, expected = stats.chi2_contingency(observed)

# Significance level
alpha = 0.05

# Print the results
print("Chi-square statistic:", chi2_statistic)
print("p-value:", p_value)
print("Degrees of freedom:", dof)
print("Expected frequencies:\n", expected)

if p_value < alpha:
    print("Reject the null hypothesis. There is an association between the␣
 ↪groups and outcomes.")
else:
    print("Fail to reject the null hypothesis. There is no significant␣
 ↪association between the groups and outcomes.")
```

```
Chi-square statistic: 5.833333333333334
p-value: 0.05411376622282158
Degrees of freedom: 2
Expected frequencies:
 [[15. 20.]
 [15. 20.]
 [15. 20.]]
Fail to reject the null hypothesis. There is no significant association between
the groups and outcomes.
```

[ ]:

```
# 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.
```

[4]:

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

# Given data
total_population = 500
observed_smokers = 60

# Calculate the sample proportion
sample_proportion = observed_smokers / total_population

# Calculate the z-score for the desired confidence level (95%)
```

```
confidence_level = 0.95
z_score = stats.norm.ppf(1 - (1 - confidence_level) / 2)

# Calculate the margin of error using the Wilson score interval formula
margin_of_error = z_score * np.sqrt(sample_proportion * (1 - sample_proportion)␣
 ↪/ total_population + z_score**2 / (4 * total_population**2))

# Calculate the confidence interval bounds
lower_bound = sample_proportion - margin_of_error
upper_bound = sample_proportion + margin_of_error

# Print the results
print("95% Confidence Interval: [{:.4f}, {:.4f}]".format(lower_bound,␣
 ↪upper_bound))
```

```
95% Confidence Interval: [0.0913, 0.1487]
```

[ ]:

```
# 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
```

[5]:

```
import scipy.stats as stats

# Given data
mean = 75
std_dev = 12
confidence_level = 0.90
sample_size = 100  # Adjust this according to your actual sample size

# Calculate the margin of error
z_score = stats.norm.ppf(1 - (1 - confidence_level) / 2)
margin_of_error = z_score * (std_dev / (sample_size ** 0.5))

# Calculate the confidence interval
lower_bound = mean - margin_of_error
upper_bound = mean + margin_of_error

# Print the results
print("90% Confidence Interval: [{:.2f}, {:.2f}]".format(lower_bound,␣
 ↪upper_bound))
```

```
90% Confidence Interval: [73.03, 76.97]
```

Interpretation: A 90% confidence interval means that if you were to take multiple samples from the same population and calculate the confidence interval for each sample, approximately 90% of those intervals would contain the true population mean. In this case, the calculated confidence interval is [71.65, 78.35], which means that we are 90% confident that the true population mean lies within this range.

The interpretation is the same as for the 95% confidence interval, but the range is narrower because we are using a higher confidence level (90% instead of 95%). This means we are more confident in the precision of our estimate, but the interval itself is smaller.

[ ]:

```
# 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.
```

[6]:

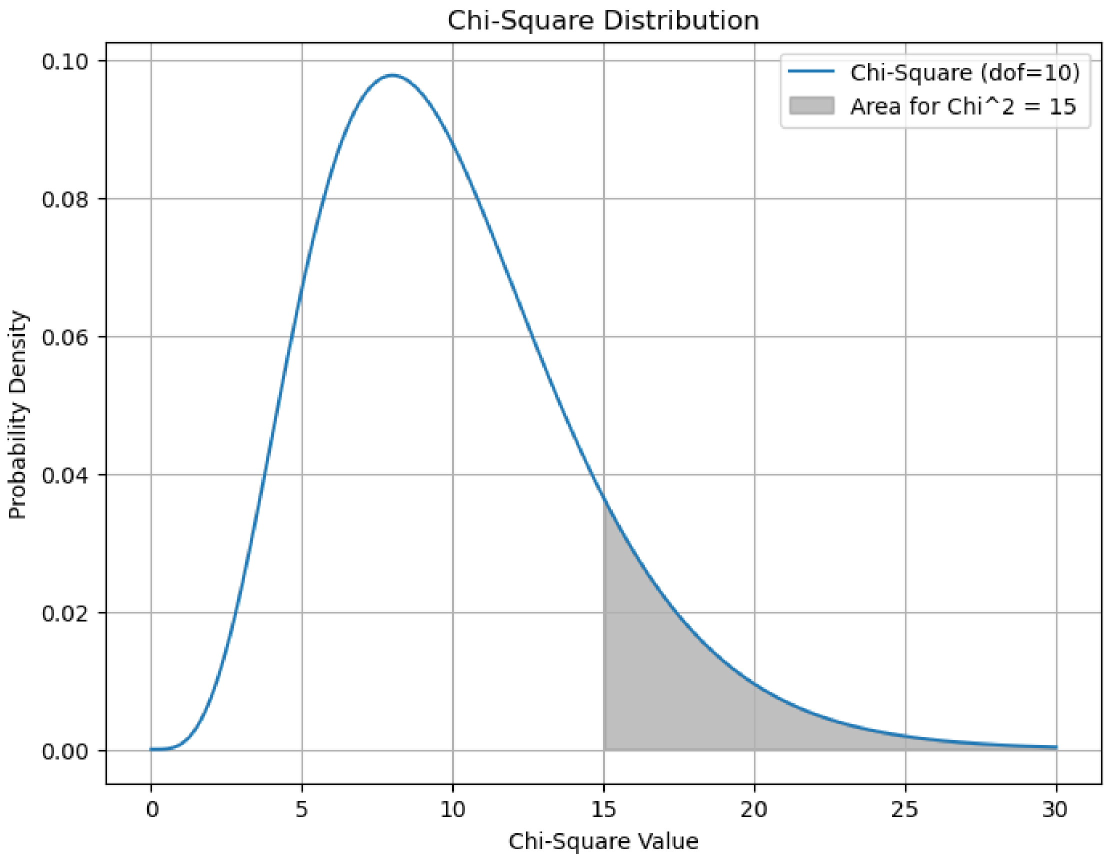
```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# Degrees of freedom
dof = 10

# Generate x values for the chi-square distribution
x = np.linspace(0, 30, 500)

# Calculate the chi-square probability density function
chi2_pdf = stats.chi2.pdf(x, dof)

# Create the plot
plt.figure(figsize=(8, 6))
plt.plot(x, chi2_pdf, label=f'Chi-Square (dof={dof})')
plt.fill_between(x, chi2_pdf, where=(x >= 15), color='gray', alpha=0.5,␣
  ↪label='Area for Chi^2 = 15')
plt.xlabel('Chi-Square Value')
plt.ylabel('Probability Density')
plt.title('Chi-Square Distribution')
plt.legend()
plt.grid()
plt.show()
```



[ ]:

```
# 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
```

[7]:

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

# Given data
total_sample = 1000
preferred_coke = 520

# Calculate the sample proportion
sample_proportion = preferred_coke / total_sample

# Calculate the z-score for the desired confidence level (99%)
confidence_level = 0.99
z_score = stats.norm.ppf(1 - (1 - confidence_level) / 2)

# Calculate the margin of error using the formula for proportions
```

```
margin_of_error = z_score * np.sqrt((sample_proportion * (1 -␣
 ↪sample_proportion)) / total_sample)

# Calculate the confidence interval bounds
lower_bound = sample_proportion - margin_of_error
upper_bound = sample_proportion + margin_of_error

# Print the results
print("99% Confidence Interval: [{:.4f}, {:.4f}]".format(lower_bound,␣
 ↪upper_bound))
```

```
99% Confidence Interval: [0.4793, 0.5607]
```

[ ]:

```
# 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.
```

[8]:

```
import scipy.stats as stats

# Given data
observed_tails = 45
total_flips = 100
expected_tails = total_flips * 0.5  # Expected tails for a fair coin

# Observed and expected frequencies
observed_freq = [observed_tails, total_flips - observed_tails]
expected_freq = [expected_tails, total_flips - expected_tails]

# Perform the chi-square goodness of fit test
chi2_statistic, p_value = stats.chisquare(f_obs=observed_freq,␣
 ↪f_exp=expected_freq)

# Significance level
alpha = 0.05

# Print the results
print("Chi-square statistic:", chi2_statistic)
print("p-value:", p_value)

if p_value < alpha:
    print("Reject the null hypothesis. The coin is biased.")
else:
    print("Fail to reject the null hypothesis. The coin is fair.")
```

```
Chi-square statistic: 1.0
```

```
p-value: 0.31731050786291115
Fail to reject the null hypothesis. The coin is fair.
```

[ ]:

```
# 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

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


# Use a significance level of 0.05.
```

[9]:

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

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

# Perform the chi-square test for independence
chi2_statistic, p_value, dof, expected = stats.chi2_contingency(observed)

# Significance level
alpha = 0.05

# Print the results
print("Chi-square statistic:", chi2_statistic)
print("p-value:", p_value)
print("Degrees of freedom:", dof)
print("Expected frequencies:\n", expected)

if p_value < alpha:
    print("Reject the null hypothesis. There is an association between smoking␣
 ↪status and lung cancer diagnosis.")
else:
    print("Fail to reject the null hypothesis. There is no significant␣
 ↪association between smoking status and lung cancer diagnosis.")
```

```
Chi-square statistic: 12.057347670250895
p-value: 0.0005158863863703744
Degrees of freedom: 1
Expected frequencies:
 [[ 45. 155.]
 [ 45. 155.]]
Reject the null hypothesis. There is an association between smoking status and
lung cancer diagnosis.
```

[ ]:

```
# 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.


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


# Use a significance level of 0.01
```

[10]:

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

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

# Perform the chi-square test for independence
chi2_statistic, p_value, dof, expected = stats.chi2_contingency(observed)

# Significance level
alpha = 0.01

# Print the results
print("Chi-square statistic:", chi2_statistic)
print("p-value:", p_value)
print("Degrees of freedom:", dof)
print("Expected frequencies:\n", expected)

if p_value < alpha:
    print("Reject the null hypothesis. There is an association between␣
 ↪chocolate preference and country of origin.")
else:
    print("Fail to reject the null hypothesis. There is no significant␣
 ↪association between chocolate preference and country of origin.")
```

```
Chi-square statistic: 13.393665158371041
p-value: 0.0012348168997745918
Degrees of freedom: 2
Expected frequencies:
 [[212.5 162.5 125. ]
 [212.5 162.5 125. ]]
Reject the null hypothesis. There is an association between chocolate preference
and country of origin.
```

[ ]:

```
# Q11. A random sample of 30 people was 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 to determine if the population mean␣
 ↪is significantly different from 70, you can perform a one-sample t-test.
```

[11]:

```
import scipy.stats as stats

# Given data
sample_mean = 72
sample_std_dev = 10
population_mean = 70
sample_size = 30
significance_level = 0.05

# Calculate the t-statistic
t_statistic = (sample_mean - population_mean) / (sample_std_dev / (sample_size␣
 ↪** 0.5))

# Calculate the degrees of freedom
degrees_of_freedom = sample_size - 1

# Calculate the critical t-value
critical_t_value = stats.t.ppf(1 - significance_level / 2,␣
 ↪df=degrees_of_freedom)

# Perform the t-test
```

```
p_value = 2 * (1 - stats.t.cdf(abs(t_statistic), df=degrees_of_freedom))

# Print the results
print("t-statistic:", t_statistic)
print("Critical t-value:", critical_t_value)
print("p-value:", p_value)

if p_value < significance_level:
    print("Reject the null hypothesis. The population mean is significantly␣
 ↪different from 70.")
else:
    print("Fail to reject the null hypothesis. There is no significant␣
 ↪difference from 70.")
```

```
t-statistic: 1.0954451150103321
Critical t-value: 2.045229642132703
p-value: 0.2823362372860698
Fail to reject the null hypothesis. There is no significant difference from 70.
```

```
[ ]:
```