Assignment

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.

Ans:To calculate the 95% confidence interval for a sample with a mean of 50 and a standard deviation of 5 using Python, you can use the following code:

python

Copy code

```
import             as


              50
             5
                   0.95
              100


                                  1                       2
                0.5






print f"95% Confidence Interval: ({lower_bound:.2f}, {upper_bound:.2f})"
```

Interpretation:

- We are 95% confident that the true population mean falls within the interval (lower_bound, upper_bound).
- In this case, if we were to repeat this process for many random samples, 95% of the calculated intervals would contain the true population mean.

Note: The `scipy.stats.norm.ppf` function is used to find the Z-score corresponding to the desired confidence level in a normal distribution. The margin of error is then calculated using this Z-score. Adjust the sample size based on the actual size of your sample.

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.
Ans:To conduct a chi-square goodness-of-fit test in Python, you can use the `scipy.stats` module. The `chisquare` function from this module can be used for this purpose.

Here's an example of how you can perform a chi-square goodness-of-fit test for the distribution of M&M colors:

python

Copy code

```
from             import
import       as



                                 50  40  30  15  15  30



                                 0.20  0.20  0.20  0.10  0.10  0.20
   sum






print f"Chi-square statistic: {chi2_stat:.4f}"
print f"P-value: {p_value:.4f}"


        0.05
if
 print "Reject the null hypothesis. The distribution of M&M colors does not match
the expected distribution."
else
 print "Fail to reject the null hypothesis. The distribution of M&M colors matches
the expected distribution."
```

In this example:

- `observed_frequencies` represent the actual counts of each color observed in the M&M bag.
- `expected_frequencies` represent the expected counts based on the specified distribution.
- The `chisquare` function is then used to perform the chi-square goodness-of-fit test, comparing the observed and expected frequencies.

Make sure to replace the example values in `observed_frequencies` with the actual counts you have for each color in your M&M bag. Adjust the `expected_frequencies` accordingly based on the expected distribution.

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

Outcome 1 20 15
Outcome 2 10 25
Outcome 3 15 20
Group B
Interpret the results of the test.
Ans:To calculate the chi-square statistic and p-value for a contingency table using Python, you can use the `scipy.stats` module. The `chi2_contingency` function from this module can be used for this purpose.

Here's an example using the provided data:

python

Copy code

```
from             import


                  20  15    10  25    15  20





print f"Chi-square statistic: {chi2_stat:.4f}"
print f"P-value: {p_value:.4f}"
```

```
        0.05
if
 print "Reject the null hypothesis. There is a significant association between
Group and Outcome."
else
 print "Fail to reject the null hypothesis. There is not enough evidence to
conclude a significant association between Group and Outcome."
```

In this example:

- `observed_data` represents the observed frequencies in the contingency table. Each sublist corresponds to a row in the table.

The `chi2_contingency` function is then used to perform the chi-square test. The function returns the chi-square statistic, p-value, degrees of freedom, and the expected frequencies.

Interpretation:

- The chi-square statistic measures the difference between the observed and expected frequencies.
- The p-value tests the null hypothesis that the categorical variables are independent (i.e., there is no association between Group and Outcome).
- If the p-value is less than the chosen significance level (e.g., 0.05), you would reject the null hypothesis, suggesting a significant association between Group and Outcome.

Make sure to replace the example values in `observed_data` with your actual data.

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.
ans:To calculate the 95% confidence interval for the true proportion of individuals in the population who smoke, you can use the formula for the confidence interval for a population proportion. The formula is given by:

Confidence Interval=(�^−��^(1−�^)�,�^+��^(1−�^)�)

Confidence Interval=(

p

$\wedge$

$-Z$

n

p

$\wedge$

$(1-$

p

$\wedge$

$)$

,

p

$\wedge$

$+Z$

n

p

$\wedge$

(1−

p

^

)

)

Where:

- �^
    - p
    - ^
    -
- is the sample proportion (the proportion of individuals who smoke).
- �
- Z is the Z-score corresponding to the desired confidence level.
- �
- n is the sample size.

Given the information:

- Sample proportion (
- �^
    - p
    - ^
    -
-) =
- $60500=0.12$
    - 500
    - 60

●

- $=0.12$ (proportion of individuals who smoke)
- Sample size (
- �
- n) = 500
- Confidence level = 0.95

Let's calculate the confidence interval using Python:

python

Copy code

```
import



                    60    500
              500
                   0.95


          1.96


                                                            1








print f"95% Confidence Interval for the Proportion of Smokers: ({lower_bound:.4f},
{upper_bound:.4f})"
```

Interpretation:

- We are 95% confident that the true proportion of individuals in the population who smoke falls within the interval (lower_bound, upper_bound).
- In this case, if we were to repeat this process for many random samples, 95% of the calculated intervals would contain the true population proportion of smokers.

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.
Ans:calculate the 90% confidence interval for a sample mean using Python, you can use the `scipy.stats` module. Specifically, you can use the `t.interval` function from this module.
Here's an example code snippet:
python

Copy code

```
import             as



       75
          12
              30



                   0.9



                                           0.5



                                   1



                               1                       2






print "90% Confidence Interval:"
```

Now, let's interpret the results. The 90% confidence interval means that if you were to take many samples and calculate a 90% confidence interval for each sample, approximately 90% of those intervals would contain the true population mean. In this case, the interval is a range of values, and you can be 90% confident that the true mean of the population falls within this range.

For example, if the result is (70, 80), it means that we are 90% confident that the true population mean is between 70 and 80.

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.
Ans:To plot the chi-square distribution with 10 degrees of freedom in Python, you can use the `scipy.stats` module. Here's an example code snippet using Matplotlib to create the plot and shade the area corresponding to a chi-square statistic of 15:
python

Copy code

```
import       as
import                   as
import             as


     10


                0  30  1000


                                         f'Chi-square Distribution (df={df})'


                     0  15  1000
                                                           'skyblue'        0.5
      'Area for chi-square = 15'


           'Chi-square Statistic'
           'Probability Density Function'
          f'Chi-square Distribution with {df} Degrees of Freedom'
```

This code uses the `stats.chi2.pdf` function to calculate the probability density function (PDF) of the chi-square distribution with 10 degrees of freedom. It then plots the distribution and shades the area corresponding to a chi-square statistic of 15.

Make sure you have Matplotlib installed (`pip install matplotlib`) before running this code.

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.
Ans:To calculate a confidence interval for the true proportion of people in the population who prefer Coke, you can use the formula for the confidence interval for a population proportion. The formula is:

Confidence Interval=(�^−��^(1−�^)�, �^+��^(1−�^)�)

Confidence Interval=(

p

^

$-z$

n

p

^

(1−

p

^

)

,

p

^

$+z$

n

p

^

$(1-$

p

^

$)$

$)$

where:

- �^
    - p
    - ^
    -
- is the sample proportion (520/1000 in this case),
- �
- n is the sample size (1000 in this case),
- �
- z is the z-score corresponding to the desired confidence level.

For a 99% confidence interval,

�

z is the critical value for the standard normal distribution, which is approximately 2.576.

Here's the Python code to calculate and print the confidence interval:

python

Copy code

```
import


              1000
                 520
                   0.99






                                                 1




                          1                       2









print f"99% Confidence Interval for the True Proportion of People Preferring Coke: {confidence_interval}"
```

Make sure to import the `scipy.stats` module at the beginning of your code.

This will print the 99% confidence interval for the true proportion of people in the population who prefer Coke.

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.
Ans:To conduct a chi-square goodness-of-fit test, you need to compare the observed frequencies with the expected frequencies and determine if there is a significant difference. For a fair coin, the expected frequency of tails in a single flip is 0.5.

Here's a step-by-step guide to performing a chi-square goodness-of-fit test in Python using the `scipy.stats` module:

python

Copy code

```
import             as


                 45
              100
                             0.5

















        0.05
```

```
print f"Chi-square Statistic: {chi2_stat}"
print f"P-value: {p_value}"


if
 print "Reject the null hypothesis. The coin is biased."
else
 print "Fail to reject the null hypothesis. There is no significant evidence of
bias."
```

In this code:

- `observed_tails` is the number of tails observed.
- `total_flips` is the total number of coin flips.
- `expected_tails_probability` is the expected probability of getting tails in a fair coin (0.5).
- `expected_tails` and `expected_heads` are the expected frequencies for tails and heads, respectively.
- `observed_frequencies` and `expected_frequencies` are arrays representing the observed and expected frequencies.
- `stats.chisquare` is used to perform the chi-square goodness-of-fit test.

The null hypothesis is that the coin is fair (not biased). If the p-value is less than the chosen significance level (0.05), you would reject the null hypothesis, suggesting evidence of bias in favor of tails. Otherwise, you would fail to reject the null hypothesis.

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

Smoker 60 140
Non-smoker 30 170
Lung Cancer: No
Use a significance level of 0.05.
Ans:To conduct a chi-square test for independence, you can use the `scipy.stats` module in Python. Here's a step-by-step guide to perform the test:

python

Copy code

```
import             as


                  60  140    30  170






        0.05



print f"Chi-square Statistic: {chi2_stat}"
print f"P-value: {p_value}"
print f"Degrees of Freedom: {dof}"
print "Expected Frequencies:"
print


if
 print "Reject the null hypothesis. There is a significant 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."
```

In this code:

- `observed_data` is a 2x2 contingency table representing the observed frequencies of the data.
- `stats.chi2_contingency` is used to perform the chi-square test for independence.
- The function returns the chi-square statistic (`chi2_stat`), p-value (`p_value`), degrees of freedom (`dof`), and expected frequencies (`expected`).

The null hypothesis is that smoking status and lung cancer diagnosis are independent. If the p-value is less than the chosen significance level (0.05), you would reject the null hypothesis, indicating a significant association between smoking status and lung cancer diagnosis. Otherwise, you would fail to reject the null hypothesis.