1. Scenario: A company wants to analyze the sales performance of its products in different regions. They have collected the following data: Region A: [10, 15, 12, 8, 14] Region B: [18, 20, 16, 22, 25] Calculate the mean sales for each region.

```
A= [10, 15, 12, 8, 14]
B= [18, 20, 16, 22, 25]
#Calculate the mean=Sum of observations/Total number of observations
Mean_A=sum(A)/len(A)
Mean_B=sum(B)/len(B)
print(Mean_A)
print(Mean_B)
```

```
11.8
20.2
```

1. Scenario: A survey is conducted to measure customer satisfaction on a scale of 1 to 5. The data collected is as follows: [4, 5, 2, 3, 5, 4, 3, 2, 4, 5] Calculate the mode of the survey responses.

```
import statistics
A=[4, 5, 2, 3, 5, 4, 3, 2, 4, 5]
#Mode means highest frequency values occur in list
mode_A = statistics.mode(A)
print(mode_A)
```

```
4
```

1. Scenario: A company wants to compare the salaries of two departments. The salary data for Department A and Department B are as follows: Department A: [5000, 6000, 5500, 7000] Department B: [4500, 5500, 5800, 6000, 5200] Calculate the median salary for each department.

```
import statistics
A= [5000, 6000, 5500, 7000]
B= [4500, 5500, 5800, 6000, 5200]
median_A=statistics.median(A)
median_B=statistics.median(B)
print(median_A)
print(median_B)
```

```
5750.0
5500
```

1. Scenario: A data analyst wants to determine the variability in the daily stock prices of a company. The data collected is as follows: [25.5, 24.8, 26.1, 25.3, 24.9] Calculate the range of the stock prices.

```
A=[25.5, 24.8, 26.1, 25.3, 24.9]
range_A = max(A) - min(A)
print(range_A)
```

```
1.3000000000000007
```

1. Scenario: A study is conducted to compare the performance of two different teaching methods. The test scores of the students in each group are as follows: Group A: [85, 90, 92, 88, 91] Group B: [82, 88, 90, 86, 87] Perform a t-test to determine if there is a significant difference in the mean scores between the two groups.

```
from scipy import stats
A=[85, 90, 92, 88, 91]
B=[82, 88, 90, 86, 87]
t_statistic, p_value = stats.ttest_ind(A, B)
alpha = 0.05

if p_value < alpha:
    print("There is a significant difference in the mean scores 
between the two groups.")
else:
    print("There is no significant difference in the mean scores 
between the two groups.")
```

```
There is no significant difference in the mean scores between the two 
groups.
```

1. Scenario: A company wants to analyze the relationship between advertising expenditure and sales. The data collected is as follows: Advertising Expenditure (in thousands): [10, 15, 12, 8, 14] Sales (in thousands): [25, 30, 28, 20, 26] Calculate the correlation coefficient between advertising expenditure and sales.

```
from scipy import stats
A=[10, 15, 12, 8, 14]
B=[25, 30, 28, 20, 26]
correlation_coefficient, p_value = stats.pearsonr(A, B)
print("Correlation Coefficient:", correlation_coefficient)
print("p-value:", p_value)
```

```
Correlation Coefficient: 0.8757511375750132
p-value: 0.05158319418821724
```

1. Scenario: A survey is conducted to measure the heights of a group of people. The data collected is as follows: [160, 170, 165, 155, 175, 180, 170] Calculate the standard deviation of the heights.

```
import statistics
A=[160, 170, 165, 155, 175, 180, 170]
standard_deviation = statistics.stdev(A)
print(standard_deviation)
```

```
8.591246929842246
```

1. Scenario: A company wants to analyze the relationship between employee tenure and job satisfaction. The data collected is as follows: Employee Tenure (in years): [2, 3, 5, 4, 6, 2,

4] Job Satisfaction (on a scale of 1 to 10): [7, 8, 6, 9, 5, 7, 6] Perform a linear regression analysis to predict job satisfaction based on employee tenure.

```
from scipy import stats
A=[2, 3, 5, 4, 6, 2, 4]
B=[7, 8, 6, 9, 5, 7, 6]
slope, intercept, r_value, p_value, std_err = stats.linregress(A, B)

print("Slope:", slope)
print("Intercept:", intercept)
print("R-squared value:", r_value**2)
print("p-value:", p_value)
print("Standard Error:", std_err)
```

```
Slope: -0.46808510638297873
Intercept: 8.595744680851062
R-squared value: 0.27099664053751393
p-value: 0.23095550952297988
Standard Error: 0.34333860006713723
```

1. Scenario: A study is conducted to compare the effectiveness of two different medications. The recovery times of the patients in each group are as follows: Medication A: [10, 12, 14, 11, 13] Medication B: [15, 17, 16, 14, 18] Perform an analysis of variance (ANOVA) to determine if there is a significant difference in the mean recovery times between the two medications.

```
from scipy import stats
A= [10, 12, 14, 11, 13]
B= [15, 17, 16, 14, 18]
f_statistic, p_value = stats.f_oneway(A, B)
alpha = 0.05

if p_value < alpha:
    print("There is a significant difference in the mean recovery times between the two medications.")
else:
    print("There is no significant difference in the mean recovery times between the two medications.")
```

```
There is a significant difference in the mean recovery times between the two medications.
```

1. Scenario: A company wants to analyze customer feedback ratings on a scale of 1 to 10. The data collected is as follows: [8, 9, 7, 6, 8, 10, 9, 8, 7, 8] Calculate the 75th percentile of the feedback ratings.

```
import numpy as np
A = [8, 9, 7, 6, 8, 10, 9, 8, 7, 8]
percentile_75th = np.percentile(A, 75)
print(percentile_75th)
```

```
8.75
```

1. Scenario: A quality control department wants to test the weight consistency of a product. The weights of a sample of products are as follows: [10.2, 9.8, 10.0, 10.5, 10.3, 10.1] Perform a hypothesis test to determine if the mean weight differs significantly from 10 grams.

```
from scipy import stats
A=[10.2, 9.8, 10.0, 10.5, 10.3, 10.1]
mean_A = np.mean(A)
alpha = 0.05  # significance level

t_statistic, p_value = stats.ttest_1samp(A, 10)

if p_value < alpha:
    print("The mean weight significantly differs from 10 grams.")
else:
    print("The mean weight does not significantly differ from 10
grams.")

The mean weight does not significantly differ from 10 grams.
```

1. Scenario: A company wants to analyze the click-through rates of two different website designs. The number of clicks for each design is as follows: Design A: [100, 120, 110, 90, 95] Design B: [80, 85, 90, 95, 100] Perform a chi-square test to determine if there is a significant difference in the click-through rates between the two designs.

```
A=[100, 120, 110, 90, 95]
B=[80, 85, 90, 95, 100]
observed = np.array([A, B])

chi2, p_value, _, _ = stats.chi2_contingency(observed)

alpha = 0.05  # significance level

if p_value < alpha:
    print("There is a significant difference in the click-through
rates between the two designs.")
else:
    print("There is no significant difference in the click-through
rates between the two designs.")

There is no significant difference in the click-through rates between
the two designs.
```

1. Scenario: A survey is conducted to measure customer satisfaction with a product on a scale of 1 to 10. The data collected is as follows: [7, 9, 6, 8, 10, 7, 8, 9, 7, 8] Calculate the 95% confidence interval for the population mean satisfaction score.

```
A=[7, 9, 6, 8, 10, 7, 8, 9, 7, 8]
confidence_level = 0.95
```

```
sample_mean = np.mean(A)
sample_standard_deviation = np.std(A, ddof=1)

sample_size = len(A)
t_critical = stats.t.ppf((1 + confidence_level) / 2, df=sample_size-1)

margin_of_error = t_critical * (sample_standard_deviation /
np.sqrt(sample_size))
confidence_interval = (sample_mean - margin_of_error, sample_mean +
margin_of_error)

print("95% Confidence Interval:", confidence_interval)

95% Confidence Interval: (7.043561120599888, 8.756438879400113)
```

1. Scenario: A company wants to analyze the effect of temperature on product performance. The data collected is as follows: Temperature (in degrees Celsius): [20, 22, 23, 19, 21] Performance (on a scale of 1 to 10): [8, 7, 9, 6, 8] Perform a simple linear regression to predict performance based on temperature.

```
A=[20, 22, 23, 19, 21]
B=[8, 7, 9, 6, 8]
slope, intercept, r_value, p_value, std_err = stats.linregress(A,B)

print("Slope:", slope)
print("Intercept:", intercept)
print("R-squared value:", r_value**2)
print("p-value:", p_value)
print("Standard Error:", std_err)

Slope: 0.5
Intercept: -2.9000000000000004
R-squared value: 0.4807692307692307
p-value: 0.19417134561205843
Standard Error: 0.30000000000000004
```

1. Scenario: A study is conducted to compare the preferences of two groups of participants. The preferences are measured on a Likert scale from 1 to 5. The data collected is as follows: Group A: [4, 3, 5, 2, 4] Group B: [3, 2, 4, 3, 3] Perform a Mann-Whitney U test to determine if there is a significant difference in the median preferences between the two groups.

```
A=[4, 3, 5, 2, 4]
B=[3, 2, 4, 3, 3]
u_statistic, p_value = stats.mannwhitneyu(A, B, alternative='two-
sided')

alpha = 0.05

if p_value < alpha:
```

```
    print("There is a significant difference in the median preferences between the two groups.")
else:
    print("There is no significant difference in the median preferences between the two groups.")
```

```
There is no significant difference in the median preferences between the two groups.
```

1. Scenario: A company wants to analyze the distribution of customer ages. The data collected is as follows: [25, 30, 35, 40, 45, 50, 55, 60, 65, 70] Calculate the interquartile range (IQR) of the ages.

```
A=[25, 30, 35, 40, 45, 50, 55, 60, 65, 70]
Q1 = np.percentile(A, 25)
Q3 = np.percentile(A, 75)

interquartile_range = Q3 - Q1
print(interquartile_range)
```

```
22.5
```

1. Scenario: A study is conducted to compare the performance of three different machine learning algorithms. The accuracy scores for each algorithm are as follows: Algorithm A: [0.85, 0.80, 0.82, 0.87, 0.83] Algorithm B: [0.78, 0.82, 0.84, 0.80, 0.79] Algorithm C: [0.90, 0.88, 0.89, 0.86, 0.87] Perform a Kruskal-Wallis test to determine if there is a significant difference in the median accuracy scores between the algorithms.

```
A=[0.85, 0.80, 0.82, 0.87, 0.83]
B=[0.78, 0.82, 0.84, 0.80, 0.79]
C=[0.90, 0.88, 0.89, 0.86, 0.87]
H, p_value = stats.kruskal(A, B, C)

alpha = 0.05  # significance level

if p_value < alpha:
    print("There is a significant difference in the median accuracy scores between the algorithms.")
else:
    print("There is no significant difference in the median accuracy scores between the algorithms.")
```

```
There is a significant difference in the median accuracy scores between the algorithms.
```

1. Scenario: A company wants to analyze the effect of price on sales. The data collected is as follows: Price (in dollars): [10, 15, 12, 8, 14] Sales: [100, 80, 90, 110, 95] Perform a simple linear regression to predict sales based on price.

```
A=[10, 15, 12, 8, 14]
B=[100, 80, 90, 110, 95]
```

```
slope, intercept, r_value, p_value, std_err = stats.linregress(A,B)

print("Slope:", slope)
print("Intercept:", intercept)
print("R-squared value:", r_value**2)
print("p-value:", p_value)
print("Standard Error:", std_err)
```

```
Slope: -3.506097560975609
Intercept: 136.3719512195122
R-squared value: 0.8064024390243901
p-value: 0.03850178234753776
Standard Error: 0.9918303504036147
```

1. Scenario: A survey is conducted to measure the satisfaction levels of customers with a new product. The data collected is as follows: [7, 8, 9, 6, 8, 7, 9, 7, 8, 7] Calculate the standard error of the mean satisfaction score.

```
A=[7, 8, 9, 6, 8, 7, 9, 7, 8, 7]
sample_size = len(A)

standard_deviation = np.std(A, ddof=1)
standard_error = standard_deviation / np.sqrt(sample_size)

print("Standard Error of the Mean:", standard_error)
```

```
Standard Error of the Mean: 0.30550504633038933
```

1. Scenario: A company wants to analyze the relationship between advertising expenditure and sales. The data collected is as follows: Advertising Expenditure (in thousands): [10, 15, 12, 8, 14] Sales (in thousands): [25, 30, 28, 20, 26] Perform a multiple regression analysis to predict sales based on advertising expenditure.

```
import statsmodels.api as sm

A=[10, 15, 12, 8, 14]
B=[25, 30, 28, 20, 26]

advertising_expenditure = np.array([A, B]).T
advertising_expenditure = sm.add_constant(advertising_expenditure)

# Create the response variable
sales = np.array([100, 120, 110, 90, 105])

# Fit the multiple regression model
model = sm.OLS(sales, advertising_expenditure)
results = model.fit()

# Print the summary of the regression analysis
print(results.summary())
```

```
                            OLS Regression Results                            
==============================================================================
========
Dep. Variable:                      y   R-squared:                       
0.966
Model:                            OLS   Adj. R-squared:                  
0.931
Method:                 Least Squares   F-statistic:                     
28.14
Date:                Mon, 10 Jul 2023   Prob (F-statistic):             
0.0343
Time:                        12:41:43   Log-Likelihood:                
-10.177
No. Observations:                   5   AIC:                             
26.35
Df Residuals:                       2   BIC:                             
25.18
Df Model:                           2                                         

Covariance Type:            nonrobust                                         

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
--------
const         33.7011     11.602      2.905      0.101     -16.219      
83.622
x1             0.6794      1.059      0.641      0.587      -3.879       
5.238
x2             2.4528      0.805      3.047      0.093      -1.011       
5.917
==============================================================================
========
Omnibus:                          nan   Durbin-Watson:                   
2.799
Prob(Omnibus):                    nan   Jarque-Bera (JB):                
0.641
Skew:                           0.253   Prob(JB):                        
0.726
Kurtosis:                       1.320   Cond. No.                         
255.
==============================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is 
correctly specified.
```

```
C:\Users\Admin\anaconda3\lib\site-packages\statsmodels\stats\
stattools.py:74: ValueWarning: omni_normtest is not valid with less
than 8 observations; 5 samples were given.
  warn("omni_normtest is not valid with less than 8 observations; %i "
```