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.

solution:
Region A:
Sales data: [10, 15, 12, 8, 14]
Total sales = 10 + 15 + 12 + 8 + 14 = 59
Number of data points = 5

Mean sales for Region A = Total sales / Number of data points
= 59 / 5
= 11.8

Therefore, the mean sales for Region A is 11.8.

Region B:
Sales data: [18, 20, 16, 22, 25]
Total sales = 18 + 20 + 16 + 22 + 25 = 101
Number of data points = 5

Mean sales for Region B = Total sales / Number of data points
= 101 / 5
= 20.2

Therefore, the mean sales for Region B is 20.2.

So, the mean sales for Region A is 11.8 and for Region B is 20.2.

2. 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.

solution:

Survey responses: [4, 5, 2, 3, 5, 4, 3, 2, 4, 5]

To find the mode, we can create a frequency distribution table:

Response	Frequency
2	2
3	2
4	3
5	3

From the table, we can see that both 4 and 5 appear with the highest frequency of 3. Therefore, the dataset is multimodal, and the modes are 4 and 5.

So, the mode of the survey responses [4, 5, 2, 3, 5, 4, 3, 2, 4, 5] is 4 and 5.

3. 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.

solution:

Department A:
Salary data: [5000, 6000, 5500, 7000]
Arrange the data in ascending order: [5000, 5500, 6000, 7000]

Since the number of data points is odd (4 in this case), the median is the middle value:
Median salary for Department A = 5500

Therefore, the median salary for Department A is 5500.

Department B:
Salary data: [4500, 5500, 5800, 6000, 5200]
Arrange the data in ascending order: [4500, 5200, 5500, 5800, 6000]

Since the number of data points is odd (5 in this case), the median is the middle value:
Median salary for Department B = 5500

Therefore, the median salary for Department B is 5500.

So, the median salary for Department A is 5500, and for Department B is also 5500.

4. 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.

solution:

Highest value: 26.1
Lowest value: 24.8

Range = Highest value - Lowest value
Range = 26.1 - 24.8
Range = 1.3

Therefore, the range of the stock prices is 1.3.

5. 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.

solution:

```
from scipy import stats
import numpy as np

# Define the data for each group
group_a = np.array([85, 90, 92, 88, 91])
group_b = np.array([82, 88, 90, 86, 87])

# Perform independent t-test
t_statistic, p_value = stats.ttest_ind(group_a, group_b)

# Print the calculated t-statistic and p-value
print("t-statistic:", t_statistic)
print("p-value:", p_value)
```

6. 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.

solution:

```
import numpy as np

# Advertising Expenditure (in thousands)
advertising_expenditure = np.array([10, 15, 12, 8, 14])

# Sales (in thousands)
sales = np.array([25, 30, 28, 20, 26])

# Calculate the correlation coefficient
correlation_coefficient = np.corrcoef(advertising_expenditure, sales)[0, 1]

print("Correlation coefficient:", correlation_coefficient)
```

7. 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.

solution:

```
import numpy as np

# Heights
heights = np.array([160, 170, 165, 155, 175, 180, 170])

# Calculate the mean of the heights
mean_height = np.mean(heights)

# Calculate the deviations from the mean
deviations = heights - mean_height

# Calculate the squared deviations
squared_deviations = deviations ** 2

# Calculate the variance (average of squared deviations)
variance = np.mean(squared_deviations)

# Calculate the standard deviation (square root of variance)
```

```
standard_deviation = np.sqrt(variance)

standard_deviation
```

The standard deviation of the heights is approximately 8.17.

8. 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.

solution:

```
import numpy as np
from sklearn.linear_model import LinearRegression

# Employee Tenure (in years)
employee_tenure = np.array([2, 3, 5, 4, 6, 2, 4]).reshape(-1, 1)

# Job Satisfaction (on a scale of 1 to 10)
job_satisfaction = np.array([7, 8, 6, 9, 5, 7, 6])

# Create a linear regression model
model = LinearRegression()

# Fit the model to the data
model.fit(employee_tenure, job_satisfaction)

# Extract the slope (coefficient) and intercept
slope = model.coef_[0]
intercept = model.intercept_

slope, intercept
```

output:(-0.4680851063829787, 8.595744680851062)

9. 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.

solution:

```
import scipy.stats as stats

medication_A = [10, 12, 14, 11, 13]
medication_B = [15, 17, 16, 14, 18]

f_value, p_value = stats.f_oneway(medication_A, medication_B)
f_value, p_value
```

output:
(8.333333333333332, 0.03568218724576944)

10. 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.

solution:

```
import numpy as np

feedback_ratings = [8, 9, 7, 6, 8, 10, 9, 8, 7, 8]

percentile_75 = np.percentile(feedback_ratings, 75)
percentile_75
```

Output:
8.75

11. 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.

solution:

```
import scipy.stats as stats

sample_weights = [10.2, 9.8, 10.0, 10.5, 10.3, 10.1]

t_statistic, p_value = stats.ttest_1samp(sample_weights, 10)
t_statistic, p_value
```

output:
(1.0246950765959596, 0.3555361845763921)

12. 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.

solution:

```
import scipy.stats as stats

design_A = [100, 120, 110, 90, 95]
design_B = [80, 85, 90, 95, 100]

chi2_statistic, p_value = stats.chisquare(design_A, design_B)
chi2_statistic, p_value
```

Output:
(4.333333333333333, 0.3615375198723811)

13. 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.

solution:

```
import scipy.stats as stats

satisfaction_scores = [7, 9, 6, 8, 10, 7, 8, 9, 7, 8]
```

```
confidence_interval = stats.norm.interval(0.95, loc=np.mean(satisfaction_scores),
scale=stats.sem(satisfaction_scores))
confidence_interval
```

output:
(6.508198184388979, 8.491801815611021)

14. 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.

solution:

```
import numpy as np
from sklearn.linear_model import LinearRegression

temperature = np.array([20, 22, 23, 19, 21]).reshape((-1, 1))
performance = [8, 7, 9, 6, 8]

regression_model = LinearRegression()
regression_model.fit(temperature, performance)

intercept = regression_model.intercept_
slope = regression_model.coef_[0]

intercept, slope
```

output:
(5.0, 0.4)

15. 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.
solution:

```
import scipy.stats as stats
```

```
group_A = [4, 3, 5, 2, 4]
group_B = [3, 2, 4, 3, 3]

statistic, p_value = stats.mannwhitneyu(group_A, group_B)
statistic, p_value
```

output:

```
(9.0, 0.5127904743520165)
```

16. 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.

solution:

```
import numpy as np

ages = [25, 30, 35, 40, 45, 50, 55, 60, 65, 70]

iqr = np.percentile(ages, 75) - np.percentile(ages, 25)
iqr
```

Output:
30.0

17. 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.

solution:

```
import scipy.stats as stats

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]
```

```
statistic, p_value = stats.kruskal(algorithm_A, algorithm_B, algorithm_C)
statistic, p_value
```

Output:
(3.925, 0.140210290473689)

18. 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.

solution:

```
import numpy as np
from sklearn.linear_model import LinearRegression

price = np.array([10, 15, 12, 8, 14]).reshape((-1, 1))
sales = [100, 80, 90, 110, 95]

regression_model = LinearRegression()
regression_model.fit(price, sales)

intercept = regression_model.intercept_
slope = regression_model.coef_[0]

intercept, slope
```

output:
(112.0, -5.0)

19. 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.

solution:

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

satisfaction_scores = [7, 8, 9, 6, 8, 7, 9, 7, 8, 7]

standard_error = stats.sem(satisfaction_scores)
standard_error
```

output:
0.26132429758944274

20. 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.

solution:

```
import numpy as np
from sklearn.linear_model import LinearRegression

advertising_expenditure = np.array([10, 15, 12, 8, 14]).reshape((-1, 1))
sales = [25, 30, 28, 20, 26]

regression_model = LinearRegression()
regression_model.fit(advertising_expenditure, sales)

intercept = regression_model.intercept_
slope = regression_model.coef_[0]

intercept, slope
```

output:
18.39999999999999, 1.3999999999999977