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.

Answer: Region A: [10, 15, 12, 8, 14] & Region B: [18, 20, 16, 22, 25]

To find the mean sales for Region A & B, we sum up all the sales values and divide by the total number of data points in Region A & B respectively (which is 5):

Mean sales for Region A = (10 + 15 + 12 + 8 + 14) / 5 = 59 / 5 = 11.8

Mean sales for Region B = (18 + 20 + 16 + 22 + 25) / 5 = 101 / 5 = 20.2

sales for each region are:

Region A: 11.8, Region B: 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.

Answer:

- The value 2 appears twice.
- The value 3 appears twice.
- The value 4 appears three times.
- The value 5 appears three times.

Both the values 4 and 5 have the highest frequency of occurrence, which is three times. Therefore, the mode of the survey responses 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.

Answer:

Median salary for Department A = (5500 + 6000) / 2 = 11500 / 2 = 5750

For Department B: [4500, 5500, 5800, 6000, 5200] Arranging the salaries in ascending order: [4500, 5200, 5500, 5800, 6000]

The median is the middle value in the sorted list, which in this case is the value in the middle position(because this list is odd):

Median salary for Department B = 5500

So, the median salary for each department is:

- Department A: 5750
- Department B: 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.

Answer:

Highest value = 26.1 Lowest value = 24.8

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

```
In [3]:
import scipy.stats as stats
group_a = [85, 90, 92, 88, 91]
group_b = [82, 88, 90, 86, 87]

t_statistic, p_value = stats.ttest_ind(group_a, group_b)
print("T-Statistic:", t_statistic)
print("P-Value:", p_value)

alpha = 0.05  # Significance level
if p_value < alpha:
    print("There is a significant difference in the mean scores between Group A and Group B.")
else:
    print("There is no significant difference in the mean scores between Group A and Group B.")
```

```
T-Statistic: 1.4312528946642733
P-Value: 0.19023970239078333
There is no significant difference in the mean scores between Group A and Group B.
```

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.

```
In [5]:
import numpy as np
import scipy.stats as stats
advertising_expenditure = [10, 15, 12, 8, 14]
sales = [25, 30, 28, 20, 26]
correlation_coefficient, _ = stats.pearsonr(advertising_expenditure, sales)
print("Correlation Coefficient:", correlation_coefficient)
#The correlation coefficient ranges from -1 to 1, where -1 represents a perfect negative correlation,
# 0 represents no correlation, and 1 represents a perfect positive correlation.
if correlation_coefficient <= -1:
    print ("perfect negative correlation")
elif correlation_coefficient <= 0:
    print("no correlation")
elif correlation_coefficient <= 1:
    print ("perfect positive correlation")
```

```
Correlation Coefficient: 0.8757511375750135
perfect positive correlation
```

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.

Answer:

Mean = (160 + 170 + 165 + 155 + 175 + 180 + 170) / 7 = 1175 / 7 = 167.86

Subtract the mean and square the differences:

(160 - 167.86)^2 = 62.22 , (170 - 167.86)^2 = 4.62, (165 - 167.86)^2 = 8.52, (155 - 167.86)^2 = 168.62 ,(175 - 167.86)^2 = 51.35 (180 - 167.86)^2 = 147.87 ,(170 - 167.86)^2 = 4.62

Average = (62.22 + 4.62 + 8.52 + 168.62 + 51.35 + 147.87 + 4.62) / 7 = 65.46

Standard Deviation = √65.46 = 8.10

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.

In [6]:

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

employee_tenure = np.array([2, 3, 5, 4, 6, 2, 4]).reshape(-1, 1)
job_satisfaction = np.array([7, 8, 6, 9, 5, 7, 6])

regression_model = LinearRegression()
regression_model.fit(employee_tenure, job_satisfaction)

coefficient = regression_model.coef_
intercept = regression_model.intercept_

print(f"Job Satisfaction = {coefficient[0]:.2f} * Tenure + {intercept:.2f}")

new_tenure = np.array([7]).reshape(-1, 1)
predicted_job_satisfaction = regression_model.predict(new_tenure)
print("Predicted Job Satisfaction:", predicted_job_satisfaction)
```

```
Job Satisfaction = -0.47 * Tenure + 8.60
Predicted Job Satisfaction: [5.31914894]
```

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.

```python
In [8]: 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)
print("f_value: ", f_value)
print("p-Value:", p_value)

alpha = 0.05  # Significance level

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.")
```

```
f_value:  16.0
p-Value: 0.003949772803445326
There is a significant difference in the mean recovery times between the two medications.
```

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.

Answer:

Sorted Data: [6, 7, 7, 8, 8, 8, 8, 9, 9, 10]

Index = (75 / 100) * (N + 1) = (0.75) * (10 + 1) = 8.25

Interpolated Value = (Value at Index 8) + (Decimal Part of Index) * (Value at Index 9) = 8 + 0.25 * (9 - 8) = 8 + 0.25 = 8.25

75th percentile of the feedback ratings is 8.25.

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.

```
In [9]:  import scipy.stats as stats

weights = [10.2, 9.8, 10.0, 10.5, 10.3, 10.1]
t_statistic, p_value = stats.ttest_1samp(weights, 10)

print("T-Statistic:", t_statistic)
print("P-Value:", p_value)

alpha = 0.05  # Significance level
if p_value < alpha:
    print("The mean weight differs significantly from 10 grams.")
else:
    print("The mean weight does not differ significantly from 10 grams.")
```

```
T-Statistic: 1.5126584522688367
P-Value: 0.19077595151110102
The mean weight does not differ significantly from 10 grams.
```

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.

Chi-squared = Σ((Observed value - Expected value)^2 / Expected value)
Chi-squared = (100 - 250)^2 / 250 + (120 - 300)^2 / 300 + (110 - 275)^2 / 275 + (90 - 225)^2 / 225 + (95 - 250)^2 / 250 + (80 - 200)^2 / 200 + (85 - 212.5)^2 / 212.5 + (90 - 225)^2 / 225 + (95 - 237.5)^2 / 237.5 + (100 - 225)^2 / 225 = 231.04

chi-squared table to find the p-value.

chi-squared statistic of 231.04 and 9 degrees of freedom(10-1) is 0.0001

The p-value is 0.0001

the p-value is less than the significance level, we reject the null hypothesis. This means that there is a significant difference in the click-through rates between the two designs.

Design A has a higher click-through rate than Design B

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.

Sample mean = (7 + 9 + 6 + 8 + 10 + 7 + 8 + 9 + 7 + 8) / 10 = 7.9

Sample standard deviation = √(Σ(data point - mean)^2 / n)
(7 - 7.9)^2 = 0.09)
(9 - 7.9)^2 = 2.89)
(6 - 7.9)^2 = 2.89)
(8 - 7.9)^2 = 0.09)
(10 - 7.9)^2 = 4.84)
(7 - 7.9)^2 = 0.09)
(8 - 7.9)^2 = 0.09)
(9 - 7.9)^2 = 2.89)
(7 - 7.9)^2 = 0.09)
(8 - 7.9)^2 = 0.09)
Sample standard deviation = √(11.84 / 10) = 1.08

The 95% confidence interval is:

Confidence interval = 7.9 ± 1.96 * 1.08 / √10 = 7.33 to 8.47 [Confidence interval = sample mean ± 1.96 * sample standard deviation / √n]

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.

In [13]:

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

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

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

coefficient = regression_model.coef_
intercept = regression_model.intercept_

print(f"Performance = {coefficient[0]:.2f} * Temperature + {intercept:.2f}")

new_temperature = np.array([24]).reshape(-1, 1)
predicted_performance = regression_model.predict(new_temperature)
print("Predicted Performance:", predicted_performance)
```

```
Performance = 0.50 * Temperature + -2.90
Predicted Performance: [9.1]
```

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.

```
In [15]:  import scipy.stats as stats

          group_a = [4, 3, 5, 2, 4]
          group_b = [3, 2, 4, 3, 3]

          u_statistic, p_value = stats.mannwhitneyu(group_a, group_b, alternative='two-sided')
          print("u-Statistic:", u_statistic)
          print("p-Value:", p_value)

          alpha = 0.05  # Significance level

          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.")

          u-Statistic: 17.0
          p-Value: 0.380836480306712
          There is no significant difference in the median preferences between the two groups.
```

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.

Answer:

Sorted Data: [25, 30, 35, 40, 45, 50, 55, 60, 65, 70]

the first quartile (Q1) and the third quartile (Q3) positions: Q1 Position = (25% / 100%) * (N + 1) = (0.25) * (10 + 1) = 2.75 = 2.75

Q3Position = (75% / 100%) * (N + 1) = (0.75) * (10 + 1) = 8.25 = 8.25

The values at the first quartile (Q1) and the third quartile (Q3):

Q1 = Value at Index 2 = 30

Q3 = Value at Index 8 = 60

Interquartile range (IQR) as the difference between Q3 and Q1: IQR = Q3 - Q1 = 60 - 30 = 30

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.

In [16]:
```python
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]

h_statistic, p_value = stats.kruskal(algorithm_a, algorithm_b, algorithm_c)
print("h-Statistic:", h_statistic)
print("p-Value:", p_value)
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.")
```

```
h-Statistic: 9.696947935368053
p-Value: 0.007840333026249539
There is a significant difference in the median accuracy scores between the algorithms.
```

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.

```
In [17]: import numpy as np
from sklearn.linear_model import LinearRegression

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

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

coefficient = regression_model.coef_
intercept = regression_model.intercept_

print(f"Sales = {coefficient[0]:.2f} * Price + {intercept:.2f}")

new_price = np.array([18]).reshape(-1, 1)
predicted_sales = regression_model.predict(new_price)
print("Predicted Sales:", predicted_sales)
```

```
Sales = -3.51 * Price + 136.37
Predicted Sales: [73.26219512]
```

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.

Answer:

Standard Error (SE) = Standard Deviation (SD) / √(Sample Size)

Sample Mean (X̄) = (Sum of all scores) / (Sample Size) = (7 + 8 + 9 + 6 + 8 + 7 + 9 + 7 + 8 + 7) / 10 = 76 / 10 = 7.6

Sample Standard Deviation (S) = √(((Sum of (scores - X̄)^2)) / (Sample Size - 1)) = √((((7-7.6)^2 + (8-7.6)^2 + (9-7.6)^2 + (6-7.6)^2 + (8-7.6)^2 + (7-7.6)^2 + (9-7.6)^2 + (7-7.6)^2 + (8-7.6)^2 + (7-7.6)^2)) / (10-1)) = √((0.36 + 0.16 + 1.96 + 1.96 + 0.16 + 0.36 + 1.96 + 0.36 + 0.16 + 0.36) / 9) = √(8.32 / 9) = √0.9244 = 0.9611

Standard Error (SE) = S / √(Sample Size) = 0.9611 / √10 = approx. 0.3041

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.

```
import numpy as np

import statsmodels.api as sm


advertising_expenditure = np.array([10, 15, 12, 8, 14])

sales = np.array([25, 30, 28, 20, 26])


advertising_expenditure = sm.add_constant(advertising_expenditure)


regression_model = sm.OLS(sales, advertising_expenditure)

regression_results = regression_model.fit()


print(regression_results.summary())
```

```
Dep. Variable:                      y   R-squared:                       0.767
Model:                            OLS   Adj. R-squared:                  0.689
Method:                 Least Squares   F-statistic:                     9.872
Date:                Mon, 17 Jul 2023   Prob (F-statistic):             0.0516
Time:                        21:09:44   Log-Likelihood:                -9.5288
No. Observations:                   5   AIC:                             23.06
Df Residuals:                       3   BIC:                             22.28
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         12.2012      4.429      2.755      0.070      -1.893      26.296
x1             1.1524      0.367      3.142      0.052      -0.015       2.320
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   1.136
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.546
Skew:                          -0.267   Prob(JB):                        0.761
Kurtosis:                       1.471   Cond. No.                         57.3
```