Scenario 1: Delivery Time Analysis for an E-commerce Company

An e-commerce company tracks delivery times (in minutes) for 15 orders:
`[25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95]`

The company wants to analyze the delivery performance using percentiles and detect if there are any unusual delivery times.

Question 1:
Calculate Q1 and Q3.

Question 2:
Find the Interquartile Range (IQR).

Question 3:
Detect Outliers using the IQR method.

Ans:

```
import numpy as np

data = [25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95]
df = pd.DataFrame(data, columns=['Values'])

# Calculate statistics
mean_value   = df['Values'].mean()
median_value = df['Values'].median()
mode_value   = df['Values'].mode()[0]   # mode() returns a Series
```

Answer to Q(1)

```
Q1 = df.describe()["Values"]["25%"]
Q2 = df.describe()["Values"]["50%"]
Q3 = df.describe()["Values"]["75%"]

MIN= df.describe()["Values"].min()
MAX= df.describe()["Values"].max()

print(df.describe())
```

Answer to Q(2)

```
IQR = Q3 - Q1

IQR1_5= 1.5 * IQR
Lesser = Q1 - IQR1_5
Higher = Q3 + IQR1_5
```

```
print("Lower=",Lower,"  Higher=",Higher)
```

Answer to Question-3

```
if Lesser < MIN:
    print("Low Outlier found")
if Higher > MAX:
    print("High Outlier found")
```

Output:

```
Lower= -10.0   Higher= 130.0

Low Outlier found
High Outlier found
```

Scenario 2: Student Score Analysis

A teacher is analyzing the mathematics scores of students in her class. The scores are:
`[45, 50, 55, 60, 60, 62, 63, 65, 90, 95]`

Question 1:
Calculate the mean, median, and mode of the scores.

Question 2:
Explain why the median might be a better representation than the mean in this case.

Ans for Question-1:

```
import numpy as np
import pandas as pd
from scipy import stats

scores = [45, 50, 55, 60, 60, 62, 63, 65, 90, 95]
dataset= pd.DataFrame(scores, columns=['Scores'])

mean   = dataset["Scores"].mean()
median = dataset["Scores"].median()
mode   = dataset["Scores"].mode()

print("Mean  :" , mean)
print("Median:" , median)
print("Mode  :" , mode[0])

OUTPUT:
Mean  : 64.5
```

```
Median: 61.0
Mode  : 60
```

Ans for Question-2

Two students scored 90 and 95, which are much higher than the rest of the class.
These high outliers pull the mean (64.5) upward, making it appear that the class performed better overall than most students actually did.

But the median (61) is a better representation of the students' typical performance because:

- It is not affected by extreme values (outliers).
- It better reflects only what the majority of students scored.

Scenario 3: Grocery Store Customer Analysis

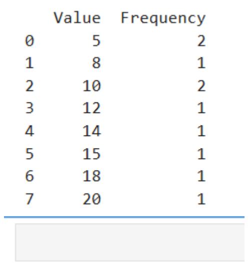
A grocery store manager tracks how many customers visit the store daily for a month:
`[5, 10, 8, 15, 20, 5, 12, 14, 10, 18]`

Question 1:
Create a frequency distribution table for this data.

Ans:

```
import pandas as pd

data = [5, 10, 8, 15, 20, 5, 12, 14, 10, 18]

# Convert data to a pandas Series
dataset = pd.Series(data)

# Use value_counts to get frequency distribution and sort by index (the values)
frequency_table = dataset.value_counts().sort_index()

# Convert to DataFrame for clearer display
frequency_df = frequency_table.reset_index()
frequency_df.columns = ['Value', 'Frequency']

print(frequency_df)
```

Output:



```
   Value  Frequency
0      5          2
1      8          1
2     10          2
3     12          1
4     14          1
5     15          1
6     18          1
7     20          1
```

Scenario 4: **Real Estate Model Analysis**

A real estate model has three variables:

- House Size
- Number of Rooms
- Number of Bathrooms

Question 1:
How can you detect multicollinearity in this model?

Ans:
We need to find both COVARIANCE

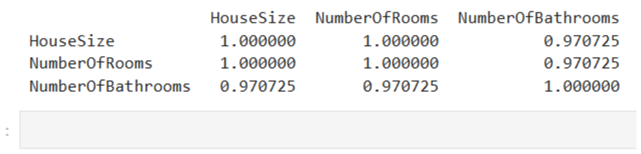
##CoVariance:

- Compute pairwise correlation coefficients between the predictors.
- High correlation (typically above 0.8 or below –0.8) between independent variables indicates potential multicollinearity.

Python Code:

```
import pandas as pd

# Example dataset
df = pd.DataFrame({
    'HouseSize': [1500, 2000, 2500, 3000, 3500],
    'NumberOfRooms': [5, 6, 7, 8, 9],
    'NumberOfBathrooms': [2, 3, 3, 4, 5]
})
print(df.corr())
```

Output:



```
                   HouseSize  NumberOfRooms  NumberOfBathrooms
HouseSize           1.000000       1.000000           0.970725
NumberOfRooms       1.000000       1.000000           0.970725
NumberOfBathrooms   0.970725       0.970725           1.000000
```

Scenario 5: Medicine Effectiveness Study

A company made a new medicine to lower blood pressure. They gave it to one group and gave a fake pill (placebo) to another group.

Question 1:
How can the company check if the new medicine works?

Ans:

The company tests the **null hypothesis** (medicine has no effect) against the **alternative hypothesis** (medicine lowers blood pressure). Statistical tests then indicate whether the new medicine works better than the placebo based on the data.

Key steps:

1. Measure blood pressure before and after treatment in both groups.
2. Statistical hypothesis testing to compare the two groups, such as:
    1. T-test for difference in means if data is normally distributed.
    2. Non-parametric tests (like Mann-Whitney U test) if data is not normal.
3. Use p-values or confidence intervals to assess whether the observed difference is statistically significant, indicating the medicine's effect is unlikely due to chance.

Scenario 6: Identifying Outliers in Sales Data

A company wants to find any unusual spikes in sales.

Question 1:
How can the company detect outliers in their sales data?

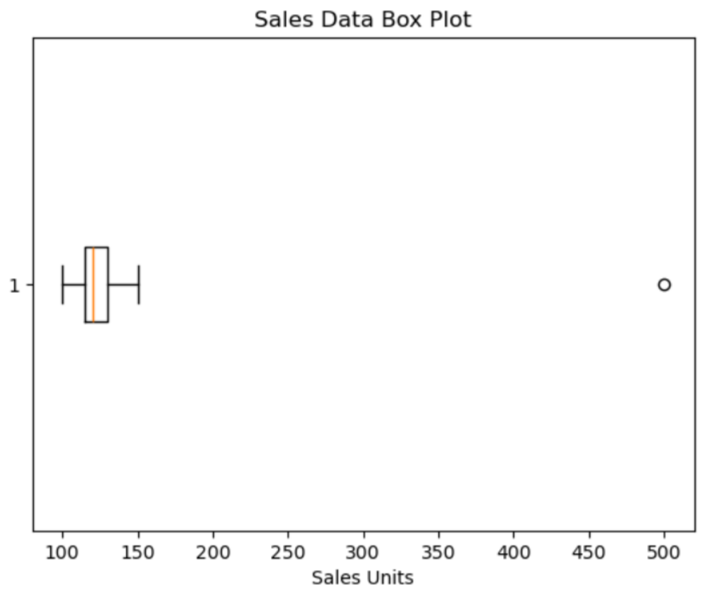
Ans:

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

# Sample sales data (weekly sales)
sales = [100, 120, 115, 130, 125, 110, 150, 115, 500, 120, 130]
sales = sorted(sales)

# ------------------------------
# 1. Detect outliers using IQR
# ------------------------------
Q1 = np.percentile(sales, 25)
Q3 = np.percentile(sales, 75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

##Finding the Sales Outlier using the Upper or Lower bound values
for x in sales:
    if x < lower_bound or x > upper_bound:
        print("Sales", x)
        iqr_outliers.append(x)

print("Upper Bound is :",upper_bound)
print("Outliers detected using IQR:", iqr_outliers)

# ------------------------------
# 2. Detect outliers using Z-score
# ------------------------------
z_scores = np.abs(stats.zscore(sales))
# print(z_scores)
zscore_outliers = [sales[i] for i in range(len(sales)) if z_scores[i] > 3]
```

```
print("Outliers detected using Z-score:", zscore_outliers)

# ------------------------------
# 3. Visualization with box plot
# ------------------------------
plt.boxplot(sales, vert=False)
plt.title("Sales Data Box Plot")
plt.xlabel("Sales Units")
plt.show()
```

Output:

```
Sales 500
Upper Bound is : 152.5
Outliers detected using IQR: [500, 500, 500]
Outliers detected using Z-score: [500]
```



Scenario 7: Understanding Customer Satisfaction

A restaurant conducted a survey to rate customer satisfaction on a scale of 1 to 5:
`[5, 4, 4, 5, 3, 4, 5, 2, 4, 3]`

Question 1:
How can the restaurant summarize the overall satisfaction?

Ans

Customer Satisfaction Dataset = [5, 4, 4, 5, 3, 4, 5, 2, 4, 3]

Mean:
= Avg[Customer Satisfaction Dataset] = 39/10
= 3.9 out of 5

Median:
Sort the data: [2, 3, 3, 4, 4, 4, 4, 5, 5, 5]
Middle two values (for 10 ratings) = 4 and 4 → Median = 4
Median shows the typical rating is 4.

Mode:
Frequency of each rating:
2 → 1 times
3 → 2 times
4 → 4 times
5 → 3 times

Mode = 4 → most customers rated 4.

4. Frequency Distribution

Rating	Count	%
2	1	10%
3	2	20%
4	4	40%
5	3	30%

Conclusion:
Average rating: 3.9 → generally good satisfaction
Median & Mode: 4 → most customers are satisfied
Frequency: Majority rated 4 or 5, indicating positive feedback.