Q1. Dataset: 5 customers with 3 attributes

Customer Data:

ID	Age	Income(K)	Education_Level
1	25	45	Bachelor
2	30	60	Master
3	35	75	PhD
4	28	50	Bachelor
5	40	85	Master

Tasks:

1. Create the data matrix (5×3)
2. Calculate pairwise Euclidean distances (age & income only)
3. Construct the 5×5 dissimilarity matrix

Q2. Medical Test Comparison:

Two tests for rare disease (1000 patients):

	Test B Positive	Test B Negative
Test A Positive	45	5
Test A Negative	10	940

Tasks:

1. Calculate Simple Matching Coefficient
2. Calculate Jaccard Coefficient

Q3. Real Estate Dataset:

House1: {Size: 1500 sqft (numeric), Type: "Apartment" (nominal), Condition: "Good" (ordinal: Poor=1, Fair=2, Good=3, Excellent=4), Garage: Yes (binary)}

House2: {Size: 1800 sqft, Type: "House", Condition: "Excellent", Garage: Yes}

1. Calculate normalized dissimilarity for each attribute
    - Size: min=1000, max=3000
    - Type: nominal (match=0, mismatch=1)
    - Condition: ordinal (normalize then absolute difference)
    - Garage: binary symmetric
2. Compute overall dissimilarity using weighted average (equal weights)

Q4. Document Vectors:

Doc1: [3, 0, 2, 1, 4] # terms: "data", "mining", "analysis", "algorithm", "python"

Doc2: [1, 2, 0, 3, 2]

Doc3: [2, 1, 1, 2, 3]

Probability Distributions (Topic Modeling):

Topic A: [0.4, 0.3, 0.2, 0.1, 0.0]

Topic B: [0.3, 0.3, 0.2, 0.1, 0.1]

Topic C: [0.5, 0.2, 0.1, 0.1, 0.1]

Tasks:

1. Calculate cosine similarity between all document pairs
2. Compute KL divergence: D_KL(TopicA || TopicB) and D_KL(TopicB || TopicA)

Q5. Dirty Customer Dataset:

ID, Name, Age, Email, Purchase_Date, Amount

1, "John Doe", 25, "john@email.com", "2023-02-15", 150.50

2, "Jane Smith", 300, "jane.email@com", "2023/13/45", 200.00

3, "Bob", -5, "bob@company.org", "2023-05-30", "one hundred"

4, "Alice Johnson", 35, "alice@", "2023-07-12", 75.25

5, , 28, "charlie@test.com", "2023-08-22", 300.00

1. Write Python code to:
    - Handle missing names (impute with "Unknown")
    - Fix invalid ages (clip to reasonable range 18-100)
    - Standardize date format (all to YYYY-MM-DD)
    - Convert amount strings to numeric

Q6. Employee Dataset:

Salaries: [45000, 52000, 48000, 75000, 82000, 68000, 92000, 55000, 62000, 150000]

Ages: [25, 32, 28, 45, 38, 41, 50, 29, 35, 42]

Performance: [3.2, 4.1, 3.8, 4.5, 4.2, 3.9, 4.8, 3.5, 4.0, 4.6] # scale 1-5

Tasks:

1. Apply three normalization techniques to salaries:
    - Min-Max normalization to [0, 1]
    - Z-score standardization
    - Decimal scaling
2. Discretize ages into bins:
    - Equal-width (3 bins)
    - Equal-frequency (3 bins)

Q7. Large E-commerce Dataset (simulated):

```
import numpy as np

np.random.seed(42)

n_samples = 10000

data = {

    'customer_id': range(n_samples),

    'age': np.random.randint(18, 70, n_samples),

    'income': np.random.normal(50000, 15000, n_samples),

    'purchases': np.random.poisson(5, n_samples),

    'segment': np.random.choice(['A', 'B', 'C', 'D'], n_samples, p=[0.1, 0.3, 0.4, 0.2])

}
```

Tasks:

1. Implement four sampling methods:
    - Simple random sampling (n=1000)
    - Stratified sampling by segment (maintain proportions)
    - Systematic sampling (every 10th record)
    - Reservoir sampling algorithm (for streaming simulation)

Q8. Iris Dataset (or synthetic data):

```
from sklearn.datasets import load_iris

iris = load_iris()
```

X = iris.data # 150 samples × 4 features

y = iris.target

Tasks:

1. Manually implement PCA:
    - Standardize the data
    - Compute covariance matrix
    - Calculate eigenvalues and eigenvectors
    - Sort by explained variance
2. Use scikit-learn PCA to verify results
3. Visualize:
    - Scree plot (variance explained)
    - Biplot (first two PCs with original features)

Q9. House Price Prediction Dataset with 10 features:

Features: Size, Bedrooms, Bathrooms, Age, Location_Score,

School_Rating, Crime_Rate, Distance_to_City,

Park_Proximity, Public_Transport

Target: Price

Tasks:

1. Apply three feature selection methods:
    - Filter: Variance threshold (remove < 0.01 variance)
    - Filter: Correlation with target & between features
    - Wrapper: Forward selection using linear regression
2. Compare selected features from each method