Q2

1. First, use the IQR method to remove outliers and generate a processed excel file. IQR Method for Outlier Removal.

   IQR Method for Outlier Removal Steps:

   1. Compute the first quartile (Q1) and the third quartile (Q3).
   2. Calculate the interquartile range (IQR): IQR=Q3−Q1IQR = Q3 - Q1
   3. Set the lower and upper bounds: Lower bound = Q1−1.5×IQRQ1 - 1.5 \times IQR Upper bound = Q3+1.5×IQRQ3 + 1.5 \times IQR
   4. Identify outliers: Any value below the lower bound or above the upper bound is considered an outlier.
   5. Replace outliers with NaN to facilitate further missing value processing.

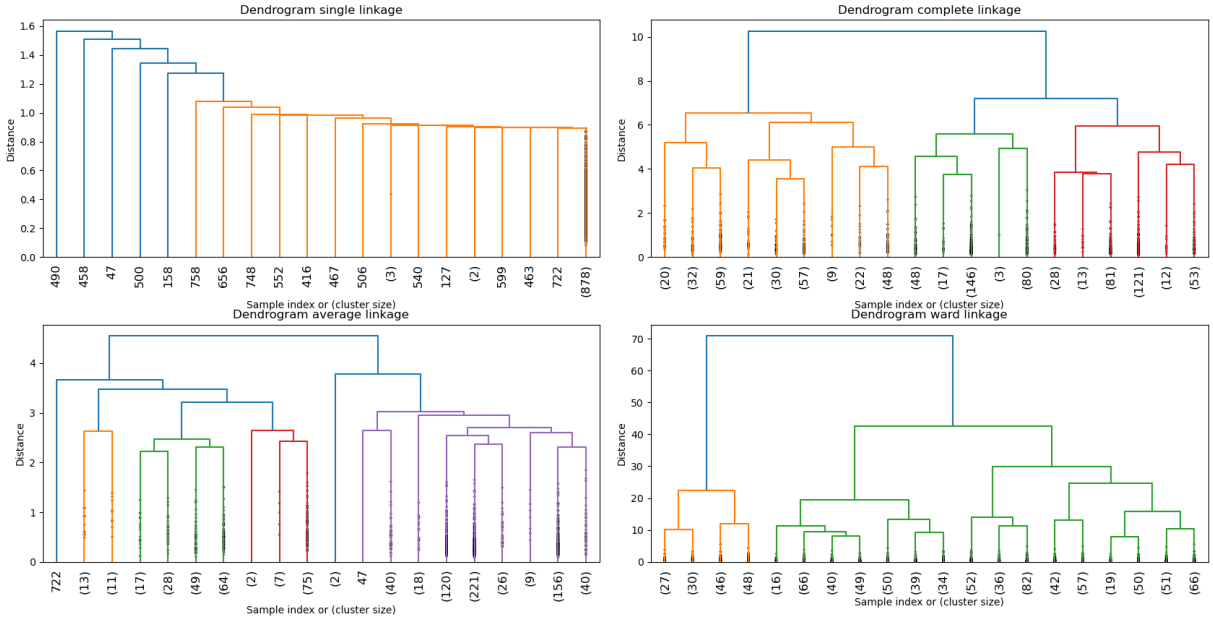
   Then fill the missing values with KNN. KNN estimates missing values based on similar data points, providing better results than simple imputation methods.

   1. Load the data after outlier removal
   2. Apply KNNImputer for imputation: choose an appropriate n_neighbors, the use Euclidean distance to find the nearest neighbors and compute the mean to fill in missing values.
   3. Save the imputed data.

   After that, we apply Z-score normalization, the processed data will have a mean of 0 and a standard deviation of 1.

2. To implement hierarchical clustering with different linkage methods, we use scipy.cluster.hierarchy to import dendrogram, linkage and fcluster.
   We define four linkage methods to compare: single, complete, average, and Ward.
   For each method:
   We record the start time to measure computational efficiency
   linkage() function computes hierarchical clustering and returns the linkage matrix Z
   Z is an (n-1)×4 matrix where n is the number of observations
   Each row contains [cluster1, cluster2, distance, sample_count]
   We store both the linkage matrix and computation time for later analysis.

3. Then we create a 2*2 grid of subplots to compare all four methods. For each method, we use the dendrogram to visualize the hierarchical clustering as a tree.



4. For the evaluation, we create a DataFrame to store performance metrices. For each method, fcluster() cuts the dendrogram to get flat clusters (here we choose 2 clusters), silhouette_score measures cluster quality (-1 to 1, higher is better). The performance Dataframe shows the computation time (in seconds) and silhouette score (cluster quality metric).

```
Performance Comparison:
   Method  Time (s)  Silhouette Score
0  single  0.009019          0.341194
D:\data process homework\homework02\Q2\2.py:61: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  performance_df = performance_df.append({

Performance Comparison:
     Method  Time (s)  Silhouette Score
0    single  0.009019          0.341194
1  complete  0.010995          0.406074
D:\data process homework\homework02\Q2\2.py:61: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  performance_df = performance_df.append({

Performance Comparison:
     Method  Time (s)  Silhouette Score
0    single  0.009019          0.341194
1  complete  0.010995          0.406074
2   average  0.009999          0.412852
D:\data process homework\homework02\Q2\2.py:61: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
  performance_df = performance_df.append({

Performance Comparison:
     Method  Time (s)  Silhouette Score
0    single  0.009019          0.341194
1  complete  0.010995          0.406074
2   average  0.009999          0.412852
3      ward  0.009999          0.432857
```

5. Analysis of Advantages and Disadvantages

   Based on the results, here's how you can analyze each method:

   Single Linkage

   - Advantages:
     - Can detect non-elliptical shapes

    - Good for identifying "chained" clusters
- Disadvantages:
    - Sensitive to noise and outliers (can cause "chaining" effect)
    - Often creates unbalanced dendrograms
- On Raisin Dataset:
    - Might show elongated clusters if raisins have continuous variation in features
    - Could be sensitive to any remaining outliers

Complete Linkage

- Advantages:
    - Less sensitive to noise and outliers than single linkage
    - Tends to create more compact clusters
- Disadvantages:
    - Can break large clusters
    - Biased towards globular clusters
- On Raisin Dataset:
    - Might perform well if raisin classes are compact and well-separated
    - Could be a good balance for medium-sized datasets

Average Linkage

- Advantages:
    - Compromise between single and complete linkage
    - Less sensitive to outliers than single linkage
- Disadvantages:
    - Computationally more intensive
    - Can still be affected by noise
- On Raisin Dataset:
    - Often a good default choice for hierarchical clustering
    - Might reveal natural groupings in raisin characteristics

Ward's Method

- Advantages:
    - Minimizes within-cluster variance
    - Tends to create clusters of similar size
    - Works well with Euclidean distance (good for standardized data)
- Disadvantages:

    - Biased towards spherical clusters
    - Sensitive to outliers
- On Raisin Dataset:
    - Likely to perform well if the raisin classes differ in multiple features
    - Good choice when you expect clusters of roughly equal size