K-means Clustering and Market Basket Analysis:

1. K-means Clustering:

a) Clusters:

(i) Initial centroids: 15, 32

* Cluster 1: [5, 10, 15, 20] (closer to 15)
* Cluster 2: [25, 30, 35] (closer to 32)

(ii) Initial centroids: 12, 30

* Cluster 1: [5, 10, 12, 20] (closer to 12)
* Cluster 2: [15, 25, 30, 35] (closer to 30)

b) SSE:

(i) 15, 32:

* Cluster 1 SSE: (5-15)^2 + (10-15)^2 + (15-15)^2 + (20-15)^2 = 20 + 25 + 0 + 25 = 70
* Cluster 2 SSE: (25-32)^2 + (30-32)^2 + (35-32)^2 = 49 + 4 + 9 = 62
* Total SSE: 70 + 62 = 132

(ii) 12, 30:

* Cluster 1 SSE: (5-12)^2 + (10-12)^2 + (12-12)^2 + (20-12)^2 = 49 + 4 + 0 + 64 = 117
* Cluster 2 SSE: (15-30)^2 + (25-30)^2 + (30-30)^2 + (35-30)^2 = 225 + 25 + 0 + 25 = 275
* Total SSE: 117 + 275 = 392

2. Market Basket Analysis and Association Rules:

* Market basket analysis uses association analysis to identify frequently purchased items together (itemsets).
* Association rules express relationships between these itemsets, like "if customer buys diapers, they also buy wipes (70% of the time)."
* This helps retailers understand customer preferences, optimize product placement, and develop targeted promotions.

3. Apriori Algorithm Example:

* Items: {bread, milk, butter, eggs}.
* Transactions: {bread, milk}, {bread, eggs}, {bread, milk, butter}, {milk, eggs}.
* Minimum support threshold = 50% (2 transactions).
* Apriori finds frequent itemsets: {bread} (100%), {milk} (75%), {bread, milk} (50%).
* Association rules: {bread} => {milk} (50%), {milk} => {bread} (50%).

4. Hierarchical Clustering Distance and Stopping:

* Distance between clusters depends on the chosen linkage method:
  + Single link: Distance between closest points in each cluster.
  + Complete link: Distance between farthest points in each cluster.
  + Average link: Average distance between all point pairs in clusters.
* Iteration stops when:
  + Desired number of clusters is reached.
  + Distance between merging clusters exceeds a threshold.
  + Change in cluster distance becomes insignificant.

5. K-means Centroid Recomputation:

* Calculate the mean of all data points assigned to each cluster.
* This new mean becomes the updated centroid for the next iteration.

6. Determining Number of Clusters:

* Silhouette coefficient: Higher values indicate better cluster separation.
* Elbow method: Plot SSE vs. number of clusters, choose the "elbow" where SSE decreases sharply.
* Domain knowledge: Use expert insights about natural groupings in the data.

7. K-means Advantages and Disadvantages:

Advantages:

* Simple and efficient.
* Easy to interpret and visualize.

Disadvantages:

* Requires pre-defining K.
* Sensitive to outliers and initial centroids.
* Assumes spherical clusters.

8. Clustering Diagram:

Imagine data points scattered on a graph. K-means draws circles (clusters) around points closest to randomly placed initial centroids. Centroids are then recomputed based on assigned points, and circles adjust accordingly. This repeats until convergence or a stopping criterion is met.

Question 9:

1. For each cluster C1, C2, and C3, calculate the mean of all data points within it. These become the new centroids for the second iteration.
2. Assign each data point to the cluster with the closest new centroid based on distance (e.g., Euclidean distance).
3. Recalculate the SSE for each cluster based on the newly assigned points and their respective centroids.
4. Sum the individual cluster SSEs to get the total SSE for the second iteration.

Example:

Assuming C1 has data points (2,2), (4,4), (6,6), calculate its new centroid as (2+4+6)/3 = 4. Similarly, find new centroids for C2 and C3. Then, assign data points to their closest new centroids and calculate SSE for each cluster based on those assignments. Finally, sum all cluster SSEs for the total SSE.

Question 10:

Here's a diagram explaining how k-means can cluster defect reports:

Imagine 20 defect reports represented as dots on a 2D plane (representing features like text similarity). Initially, 5 random centroids (C1 to C5) are placed. Each report is assigned to the closest centroid, forming 5 clusters.

Iteration 1:

1. Calculate the centroid of each cluster by averaging the features of its member reports (e.g., average text similarity).
2. Reassign each report to the closest updated centroid (not the initial random ones).
3. Repeat steps 1 and 2 until cluster assignments stabilize (convergence) or a predefined stopping criterion is met.

New Defect:

* When a new defect report arrives, calculate its feature values (text similarity).
* Assign it to the cluster with the closest centroid (based on feature similarity).
* If the new report doesn't fit well within any existing cluster, consider creating a new cluster or revising the existing ones based on further analysis.