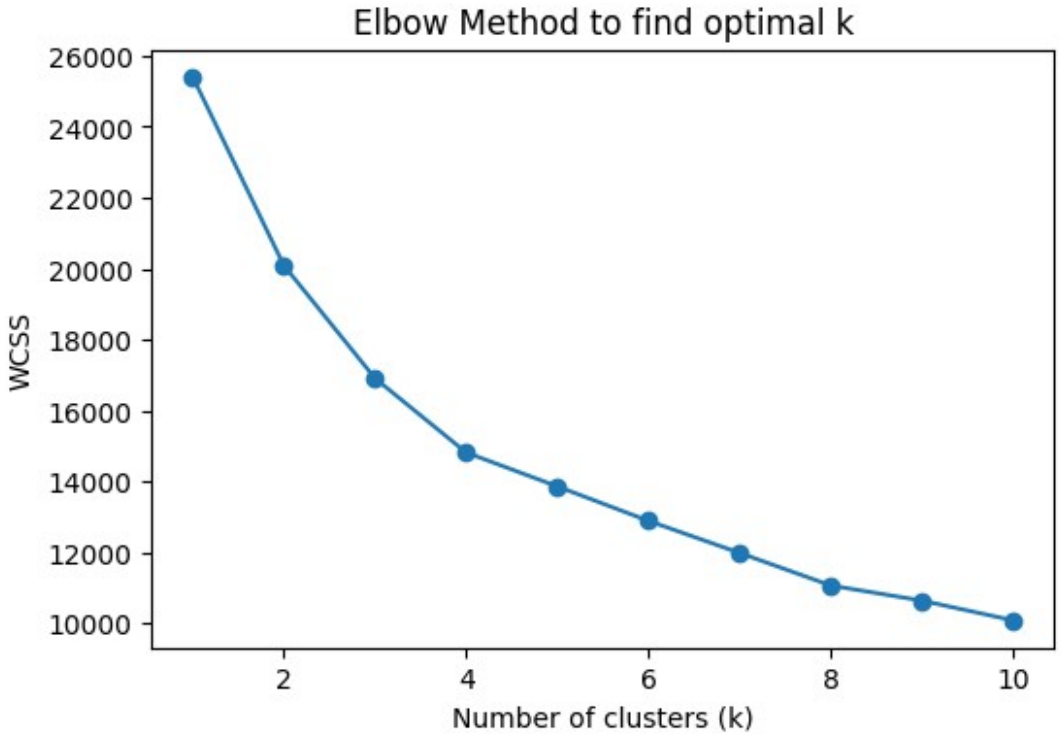
```python
# 1 Import libraries
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

# 2 Load dataset with correct encoding
df = pd.read_csv("sales_data_sample.csv", encoding='ISO-8859-1')
print(df.head())
```

```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER
SALES  \
0        10107               30      95.70                2  2871.00

1        10121               34      81.35                5  2765.90

2        10134               41      94.74                2  3884.34

3        10145               45      83.26                6  3746.70

4        10159               49     100.00               14  5205.27


         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                    ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME
DEALSIZE
0      10022     USA       NaN              Yu             Kwai
Small
1      51100  France      EMEA         Henriot             Paul
Small
2      75508  France      EMEA        Da Cunha            Daniel
Medium
3      90003     USA       NaN           Young             Julie
Medium
4        NaN     USA       NaN           Brown             Julie
Medium
```

```
[5 rows x 25 columns]
```

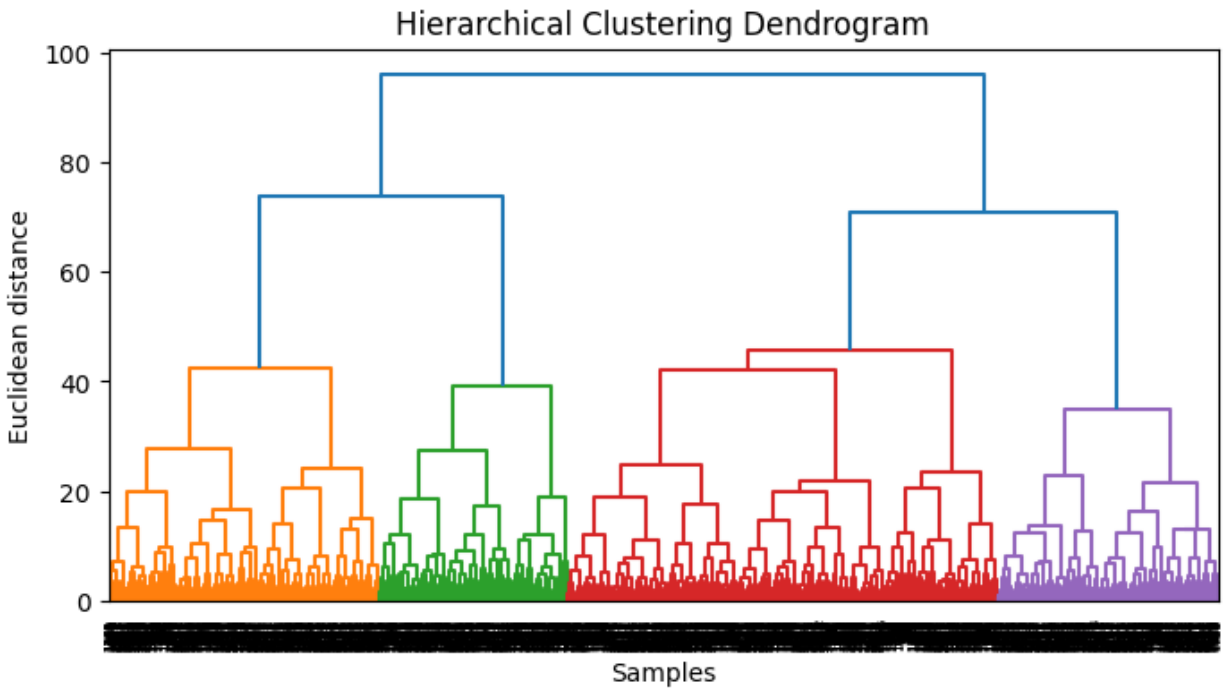
```python
# 3 Select numeric columns for clustering
numeric_cols = df.select_dtypes(include='number').columns
X = df[numeric_cols]

# 4 Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 5 Determine number of clusters using Elbow method
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot elbow
plt.figure(figsize=(6,4))
plt.plot(range(1,11), wcss, marker='o')
plt.title("Elbow Method to find optimal k")
plt.xlabel("Number of clusters (k)")
plt.ylabel("WCSS")
plt.show()
```



```
# 6 K-Means clustering
k = 3  # choose based on elbow
kmeans = KMeans(n_clusters=k, random_state=42)
df['KMeans_Cluster'] = kmeans.fit_predict(X_scaled)
print("\nK-Means cluster assignments:\n",
df[['KMeans_Cluster']].head())
```

```
K-Means cluster assignments:
    KMeans_Cluster
0                1
1                2
2                1
3                1
4                1
```

```
# 7 Hierarchical Clustering (Dendrogram)
linked = linkage(X_scaled, method='ward')
plt.figure(figsize=(8,4))
dendrogram(linked, orientation='top', distance_sort='descending',
show_leaf_counts=False)
plt.title("Hierarchical Clustering Dendrogram")
plt.xlabel("Samples")
plt.ylabel("Euclidean distance")
plt.show()
```




```
# 8 Assign clusters from hierarchical clustering
df['Hier_Cluster'] = fcluster(linked, 3, criterion='maxclust')
```

```
print("\nHierarchical cluster assignments:\n",
df[['Hier_Cluster']].head())
```

```
Hierarchical cluster assignments:
    Hier_Cluster
0              3
1              3
2              1
3              1
4              1
```