```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster
```

```
df = pd.read_csv("/content/sales_data_sample.csv" ,encoding='latin1')   # Change path if needed
print("✅ Dataset Loaded Successfully")
print(df.head())
print("\nColumns:", df.columns)
```

```
✅ Dataset Loaded Successfully
   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]

Columns: Index(['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER',
       'SALES', 'ORDERDATE', 'STATUS', 'QTR_ID', 'MONTH_ID', 'YEAR_ID',
       'PRODUCTLINE', 'MSRP', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE',
       'ADDRESSLINE1', 'ADDRESSLINE2', 'CITY', 'STATE', 'POSTALCODE',
       'COUNTRY', 'TERRITORY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME',
       'DEALSIZE'],
      dtype='object')
```

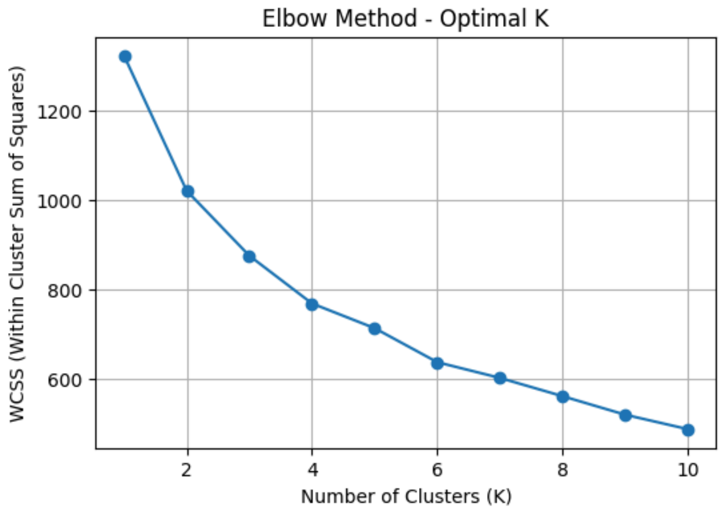
```
df = df.dropna()

# Step 4: Select numeric features for clustering
X = df.select_dtypes(include=['float64', 'int64'])

print("\n✅ Numerical Columns used for Clustering:")
print(X.columns)

# Step 5: Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
✅ Numerical Columns used for Clustering:
Index(['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER',
       'SALES', 'QTR_ID', 'MONTH_ID', 'YEAR_ID', 'MSRP'],
      dtype='object')
```

```
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42, n_init='auto')
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)
```

```
plt.figure(figsize=(6,4))
plt.plot(range(1, 11), wcss, marker='o')
plt.title("Elbow Method - Optimal K")
plt.xlabel("Number of Clusters (K)")
plt.ylabel("WCSS (Within Cluster Sum of Squares)")
```

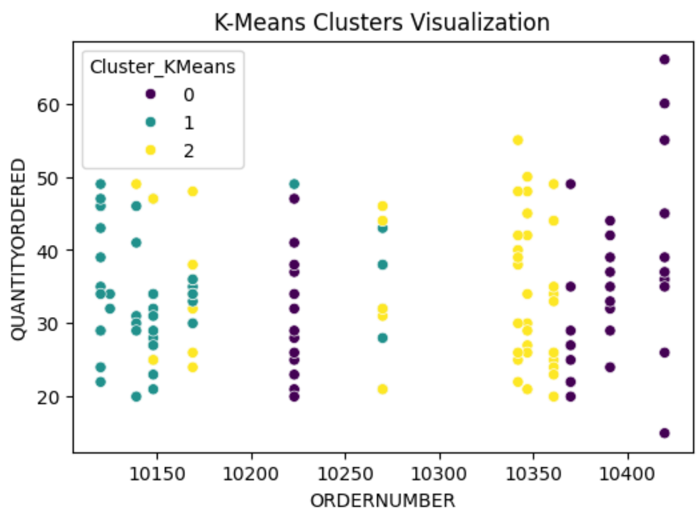
```
plt.grid()
plt.show()
```




```
k = 3   # Change based on elbow result
kmeans = KMeans(n_clusters=k, random_state=42, n_init='auto')
df["Cluster_KMeans"] = kmeans.fit_predict(X_scaled)

print("\n✅ KMeans Clustering Completed")
print(df["Cluster_KMeans"].value_counts())
```

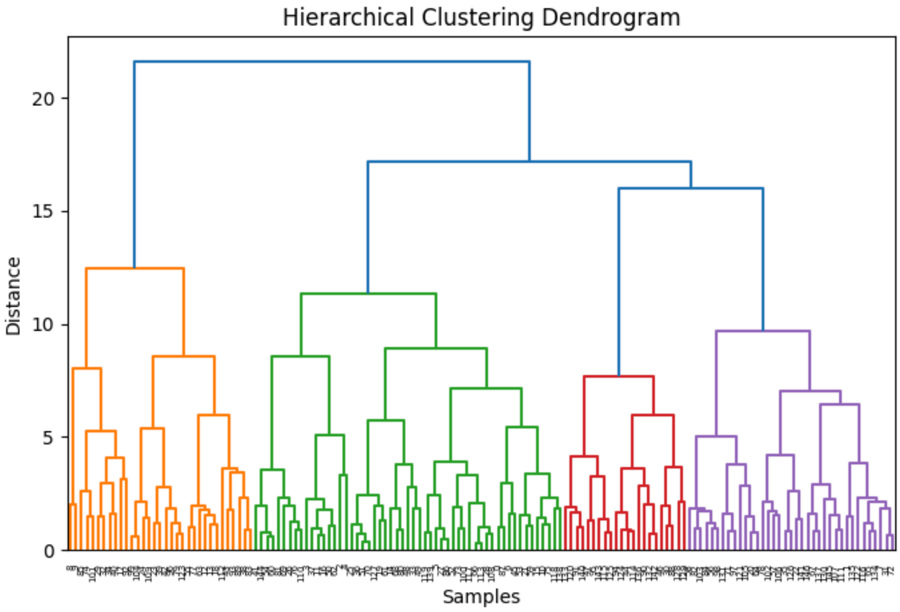
```
✅ KMeans Clustering Completed
Cluster_KMeans
2    53
1    48
0    46
Name: count, dtype: int64
```

```
# Step 8: Visualize clusters (first 2 features)
plt.figure(figsize=(6,4))
sns.scatterplot(
    x=X.iloc[:,0], y=X.iloc[:,1],
    hue=df["Cluster_KMeans"], palette='viridis'
)
plt.title("K-Means Clusters Visualization")
plt.xlabel(X.columns[0])
plt.ylabel(X.columns[1])
plt.show()
```




```
linked = linkage(X_scaled, method='ward')

plt.figure(figsize=(8,5))
dendrogram(linked)
plt.title("Hierarchical Clustering Dendrogram")
plt.xlabel("Samples")
plt.ylabel("Distance")
plt.show()
```

```
# Assign clusters (example: 3)
df["Cluster_Hier"] = fcluster(linked, k, criterion='maxclust')
print("\n✅ Hierarchical Clustering Completed")
print(df["Cluster_Hier"].value_counts())
```




```
✅ Hierarchical Clustering Completed
Cluster_Hier
3    59
2    55
1    33
Name: count, dtype: int64
```

Start coding or generate with AI.