```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.metrics import silhouette_score
```

```
df = pd.read_csv('Wine_clust.csv')
df.head()
df.info()
df.describe()
```

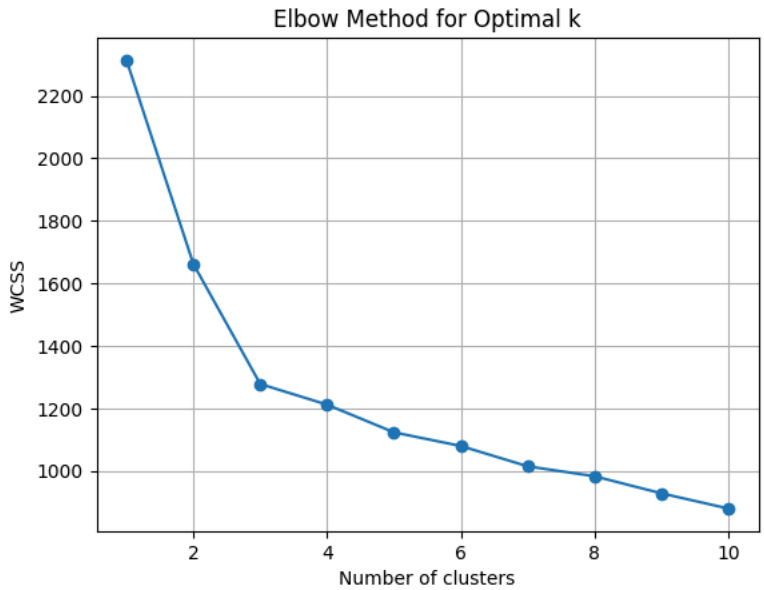
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Alcohol               178 non-null    float64
 1   Malic_Acid            178 non-null    float64
 2   Ash                   178 non-null    float64
 3   Ash_Alcanity          178 non-null    float64
 4   Magnesium             178 non-null    int64
 5   Total_Phenols         178 non-null    float64
 6   Flavanoids            178 non-null    float64
 7   Nonflavanoid_Phenols  178 non-null    float64
 8   Proanthocyanins       178 non-null    float64
 9   Color_Intensity       178 non-null    float64
 10  Hue                   178 non-null    float64
 11  OD280                 178 non-null    float64
 12  Proline               178 non-null    int64
dtypes: float64(11), int64(2)
memory usage: 18.2 KB
```

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Col
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	

```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
```

```
wcss = []                                   #KMEAN
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_data)
    wcss.append(kmeans.inertia_)

# Plotting the Elbow Curve
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()
```




```
for i in range(2, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    labels = kmeans.fit_predict(scaled_data)
    score = silhouette_score(scaled_data, labels)
    print(f"Silhouette Score for k={i}: {score}")
```

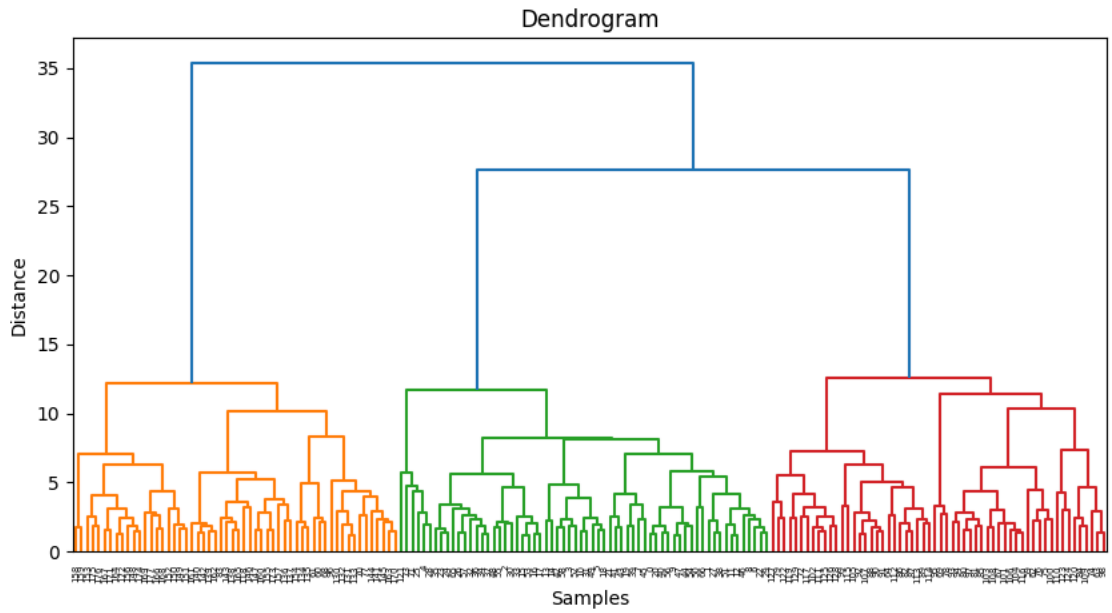
```
Silhouette Score for k=2: 0.2650328591008738
Silhouette Score for k=3: 0.2848589191898987
Silhouette Score for k=4: 0.25422758316007776
Silhouette Score for k=5: 0.18362105107698137
Silhouette Score for k=6: 0.16899191019013057
Silhouette Score for k=7: 0.1726015561094921
Silhouette Score for k=8: 0.16250411307671142
Silhouette Score for k=9: 0.1738739334545086
Silhouette Score for k=10: 0.13956723664297546
```

```
kmeans = KMeans(n_clusters=3, random_state=42)
df['KMeans_Cluster'] = kmeans.fit_predict(scaled_data)
df.head()
```

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
linked = linkage(scaled_data, method='ward')              #HIERARCHICAL
plt.figure(figsize=(10, 5))
dendrogram(linked)
plt.title('Dendrogram')
plt.xlabel('Samples')
plt.ylabel('Distance')
plt.show()
```




```
hc = AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='ward')
df['Hierarchical_Cluster'] = hc.fit_predict(scaled_data)
df.head()
```

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
db = DBSCAN(eps=2, min_samples=5)  #DBSCAN
df['DBSCAN_Cluster'] = db.fit_predict(scaled_data)
df['DBSCAN_Cluster'].value_counts()
```

DBSCAN_Cluster	count
-1	85
0	66
4	9
1	8
2	5
3	5

dtype: int64

```
df[['KMeans_Cluster', 'Hierarchical_Cluster', 'DBSCAN_Cluster']].head(10)
```

	KMeans_Cluster	Hierarchical_Cluster	DBSCAN_Cluster
0	2	2	0
1	2	2	0
2	2	2	0
3	2	2	0
4	2	2	-1
5	2	2	0
6	2	2	0