```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.datasets import load_wine
```

```
data=load_wine()
df=pd.DataFrame(data=data.data, columns=data.feature_names)
```

```
df
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flav
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95.0	1.68	
174	13.40	3.91	2.48	23.0	102.0	1.80	
175	13.27	4.28	2.26	20.0	120.0	1.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	
177	14.13	4.10	2.74	24.5	96.0	2.05	

178 rows × 13 columns

Next steps: Generate code with df | New interactive sheet

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

```
df_scaled= scaler.fit_transform(df)
```

```
df_scaled
```

```
array([[ 1.51861254, -0.5622498 ,  0.23205254, ...,  0.36217728,
         1.84791957,  1.01300893],
       [ 0.24628963, -0.49941338, -0.82799632, ...,  0.40605066,
         1.1134493 ,  0.96524152],
```

```
       [ 0.19687903,  0.02123125,  1.10933436, ...,  0.31830389,
         0.78858745,  1.39514818],
       ...,
       [ 0.33275817,  1.74474449, -0.38935541, ..., -1.61212515,
        -1.48544548,  0.28057537],
       [ 0.20923168,  0.22769377,  0.01273209, ..., -1.56825176,
        -1.40069891,  0.29649784],
       [ 1.39508604,  1.58316512,  1.36520822, ..., -1.52437837,
        -1.42894777, -0.59516041]])
```

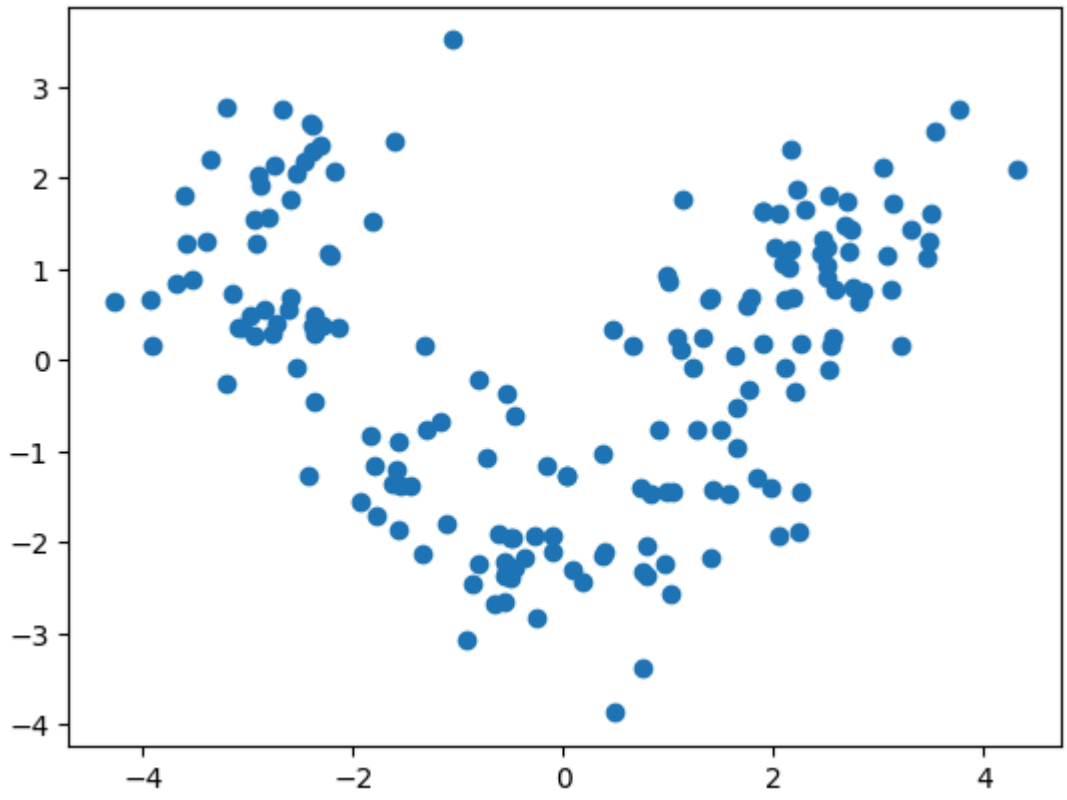
```
from sklearn.decomposition import PCA
pca=PCA(n_components=2)
pca_scaled = pca.fit_transform(df_scaled)
```

```
pca_scaled.shape
```

```
(178, 2)
```

```
plt.scatter(pca_scaled[:, 0], pca_scaled[:, 1])
```

```
<matplotlib.collections.PathCollection at 0x79418172b830>
```



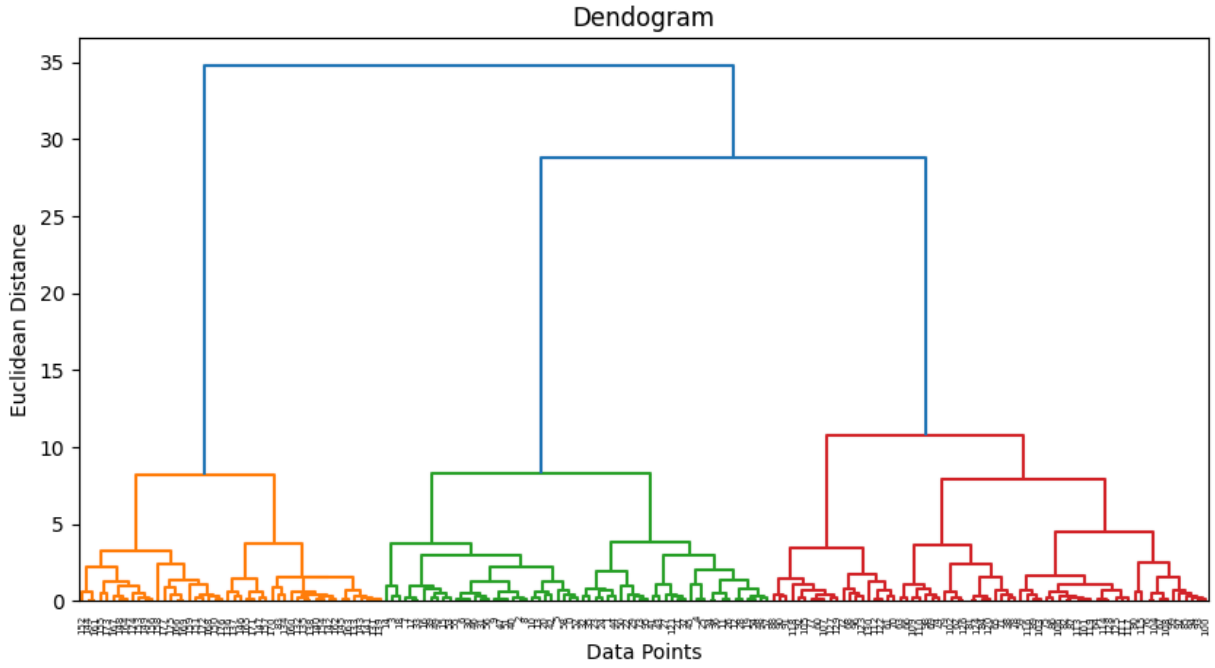

```
import scipy.cluster.hierarchy as sc
```

```
plt.figure(figsize=(10,5))
plt.title('Dendogram')


# create a Dendogram
sc.dendrogram(sc.linkage(pca_scaled, method='ward'))
```

```
plt.title('Dendogram')
plt.xlabel('Data Points')
plt.ylabel('Euclidean Distance')
```

```
Text(0, 0.5, 'Euclidean Distance')
```



Hence the n_cluster value is 3

```
# silhoutte score
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

for k in range(2,11):
    kmeans = KMeans(n_clusters=k, init='k-means++')
    labels = kmeans.fit_predict(pca_scaled)
    score = silhouette_score(pca_scaled, labels)
    print(f"for k={k} silhouette score is {score}")
```

```
for k=2 silhouette score is 0.4750102141135033
for k=3 silhouette score is 0.5609305721582905
for k=4 silhouette score is 0.4855976780800213
for k=5 silhouette score is 0.4336819174782484
for k=6 silhouette score is 0.4244059902675462
for k=7 silhouette score is 0.42251924243479017
for k=8 silhouette score is 0.3777792659691172
for k=9 silhouette score is 0.380704142674008
for k=10 silhouette score is 0.40269911475418657
```

```
# Agglomirative Clustering
from sklearn.cluster import AgglomerativeClustering
```

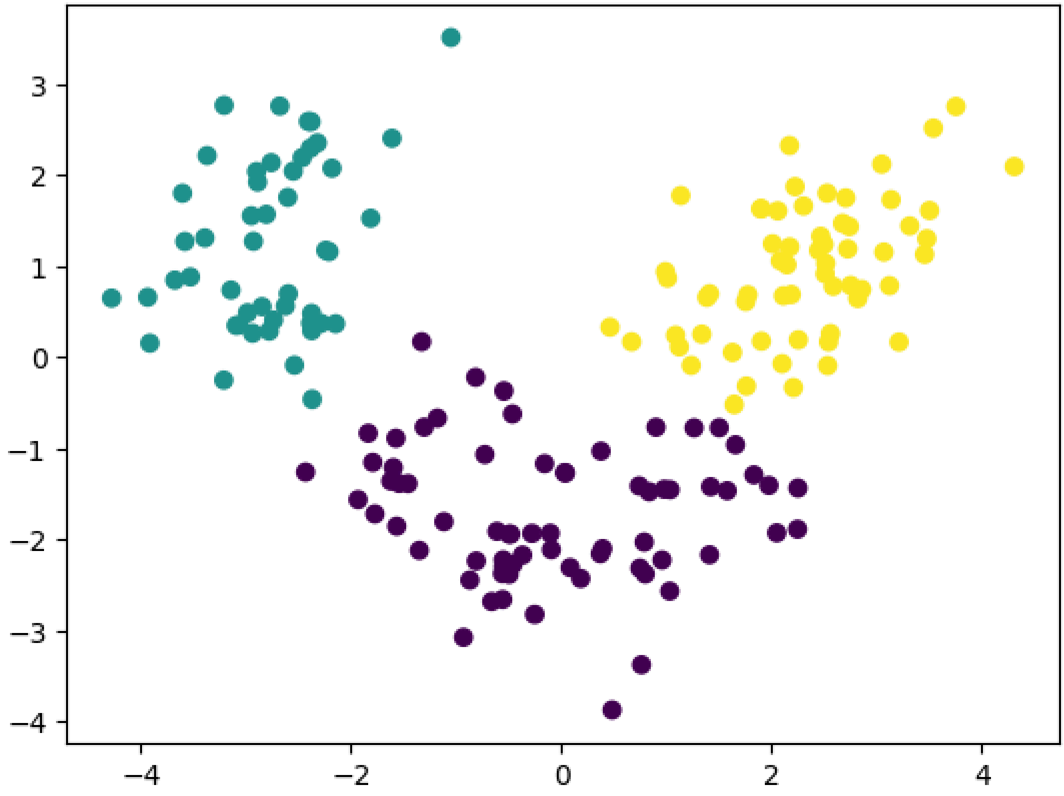
```
cluster=AgglomerativeClustering(n_clusters=3, metric='euclidean', linkage='warc
```

```
cluster_labels = cluster.fit_predict(pca_scaled)
print(cluster_labels)
```

```
[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

```
plt.scatter(pca_scaled[:, 0], pca_scaled[:, 1], c=cluster_labels)
```

```
<matplotlib.collections.PathCollection at 0x79417d9759d0>
```

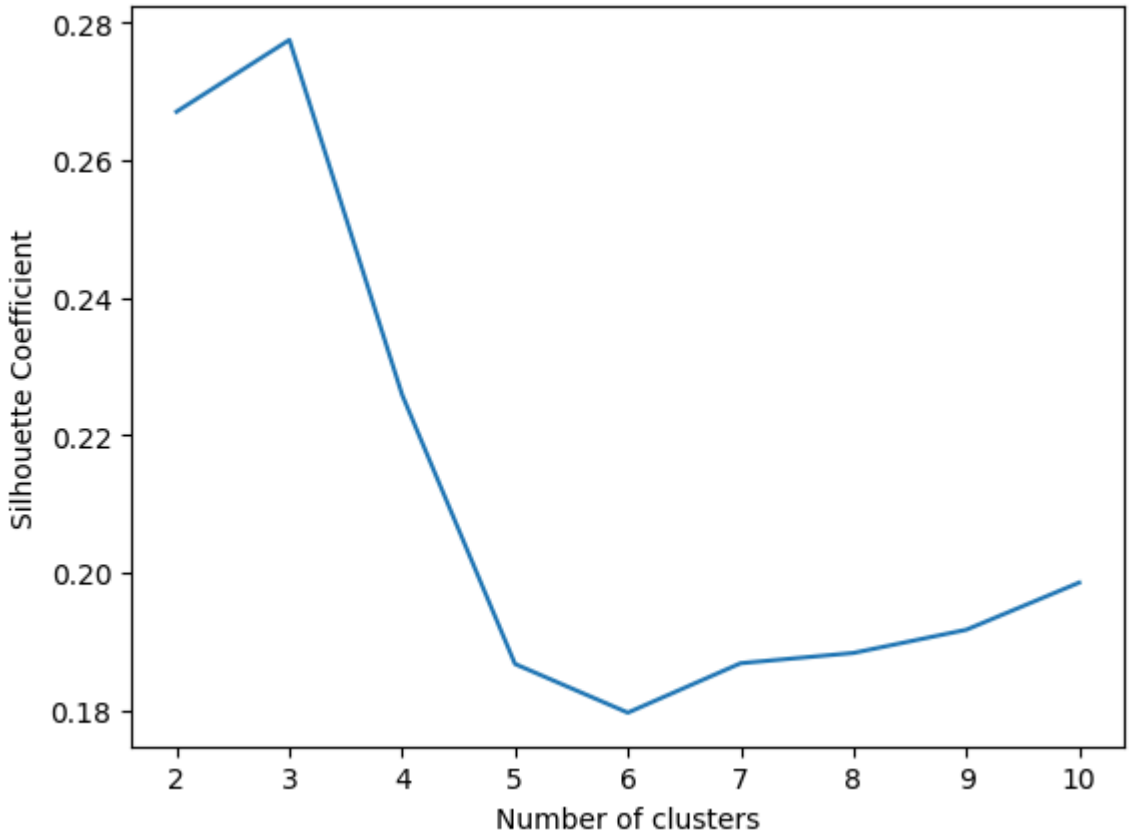



```
silhouette_coeff=[]
for k in range(2,11):
    agglo = AgglomerativeClustering(n_clusters=k, metric='euclidean', linkag
    agglo_labels = agglo.fit_predict(df_scaled)
    score = silhouette_score(df_scaled, agglo_labels)
    silhouette_coeff.append(score)
    print(f"for k={k} silhouette score is {score}")
```

```
for k=2 silhouette score is 0.2670131771272231
for k=3 silhouette score is 0.2774439826952266
for k=4 silhouette score is 0.22583665933475802
for k=5 silhouette score is 0.18674235566758707
for k=6 silhouette score is 0.17966642854438503
for k=7 silhouette score is 0.1868534256022694
for k=8 silhouette score is 0.18834697102837822
for k=9 silhouette score is 0.1917169293227209
for k=10 silhouette score is 0.19856750165505588
```

```
plt.plot(range(2,11), silhouette_coeff)
plt.xticks(range(2,11))
```

```
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Coefficient')
plt.show()
```



Start coding or generate with AI.