Importing Necessary libaries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.datasets import load_wine
```

Loading Dataset and Data Preprocessing

```
wine_data = load_wine(as_frame = True)
wine_data = wine_data.frame
wine_data.head()
```

ol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids
23	1.71	2.43	15.6	127.0	2.80	3.06
20	1.78	2.14	11.2	100.0	2.65	2.76
16	2.36	2.67	18.6	101.0	2.80	3.24
37	1.95	2.50	16.8	113.0	3.85	3.49
24	2.59	2.87	21.0	118.0	2.80	2.69

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

```
wine_data.drop('target',axis=1,inplace=True)
wine_data.info()
```

```
<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   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 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/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
dtypes: float64(13)
memory usage: 18.2 KB
```

Standardizing the data

```
scaler = StandardScaler()
scaled_wine_data = scaler.fit_transform(wine_data)


pca = PCA()
wine_pca = pca.fit_transform(scaled_wine_data)


wine_pca = pd.DataFrame(wine_pca,columns=wine_data.columns)
wine_pca.head()
```

flavan
0.59
0.05
0.42
-0.38
0.44

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

Applying PCA on standardized data

```
pca = PCA(n_components=2)
wine_pca = pca.fit_transform(scaled_wine_data)
wine_pca = pd.DataFrame(wine_pca,columns=['PC1','PC2'])
wine_pca.head()
```

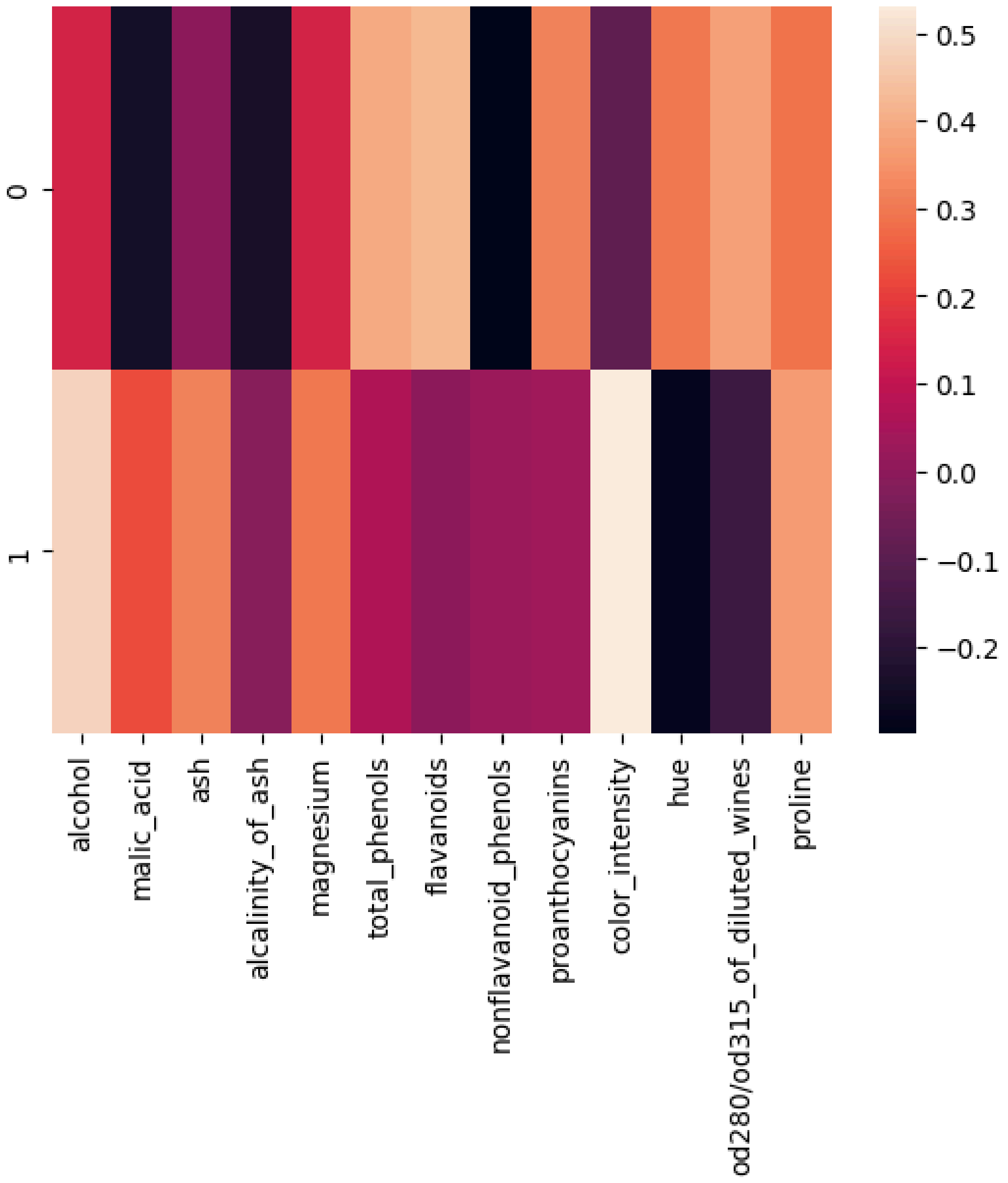
	PC1	PC2
0	3.316751	1.443463
1	2.209465	-0.333393
2	2.516740	1.031151
3	3.757066	2.756372
4	1.008908	0.869831

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

```
pca.components_
```

```
array([[ 0.1443294 , -0.24518758, -0.00205106, -0.23932041,  0.14199204,
         0.39466085,  0.4229343 , -0.2985331 ,  0.31342949, -0.0886167 ,
         0.29671456,  0.37616741,  0.28675223],
       [ 0.48365155,  0.22493093,  0.31606881, -0.0105905 ,  0.299634  ,
         0.06503951, -0.00335981,  0.02877949,  0.03930172,  0.52999567,
        -0.27923515, -0.16449619,  0.36490283]])
```

```
component_df = pd.DataFrame(pca.components_,columns=wine_data.columns)
sns.heatmap(component_df)
plt.show()
```



Start coding or generate with AI.