Wine Clustering

Wyatt Rasmussen

DSC 680

Summary:

For this project the goal is to take in the wines in the dataset and cluster them into groups that are similar based on the chemical makeup of the wine. This will require unsupervised learning as we do not have an answer within the data of the clusters that these wines should belong to. To do this project I will be using a clustering algorithm and adding that as a column in the data. Then the data will be import over to Tableau to do visualizations with the clusters.

Imports

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import plotly.express as px
```

In [2]:
```
wines = pd.read_csv('data/wine-clustering.csv')
```

Understanding the Data

In [3]:
```
shape = wines.shape

print('There are {} rows'.format(shape[0]))
print('There are {} columns'.format(shape[1]))
```

```
There are 178 rows
There are 13 columns
```

In [4]:
```
wines.isna().sum()
```

Out[4]:
```
Alcohol                 0
Malic_Acid              0
Ash                     0
Ash_Alcanity            0
Magnesium               0
Total_Phenols           0
Flavanoids              0
Nonflavanoid_Phenols    0
Proanthocyanins         0
Color_Intensity         0
Hue                     0
OD280                   0
```

```
Proline                 0
dtype: int64
```

In [5]:
```
wines.describe()
```

Out[5]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids
count	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
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000

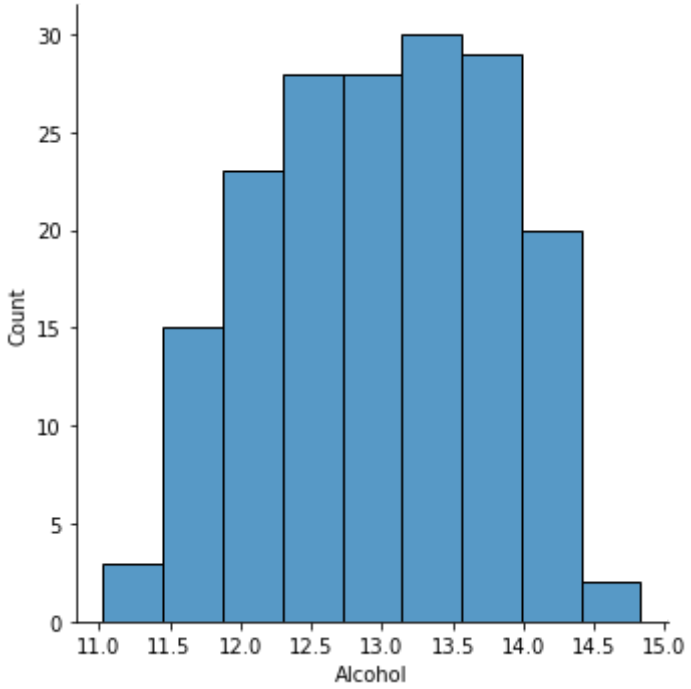
In [6]:
```
wines.dtypes
```

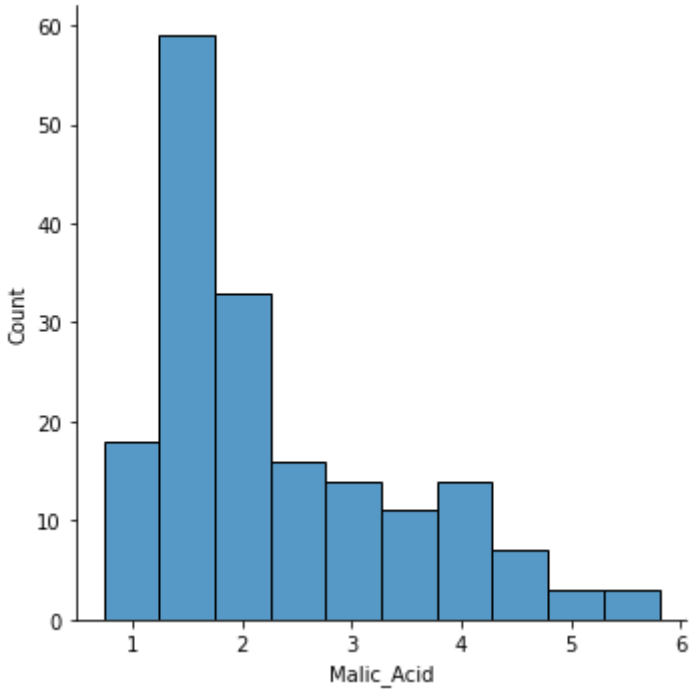
Out[6]:
```
Alcohol                 float64
Malic_Acid              float64
Ash                     float64
Ash_Alcanity            float64
Magnesium                 int64
Total_Phenols           float64
Flavanoids              float64
Nonflavanoid_Phenols    float64
Proanthocyanins         float64
Color_Intensity         float64
Hue                     float64
OD280                   float64
Proline                   int64
dtype: object
```

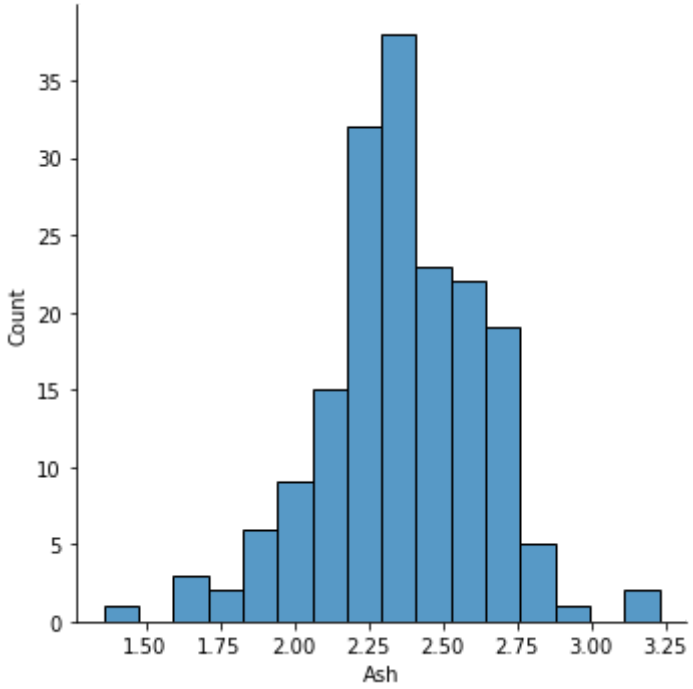
In [7]:
```
columns = wines.columns
```

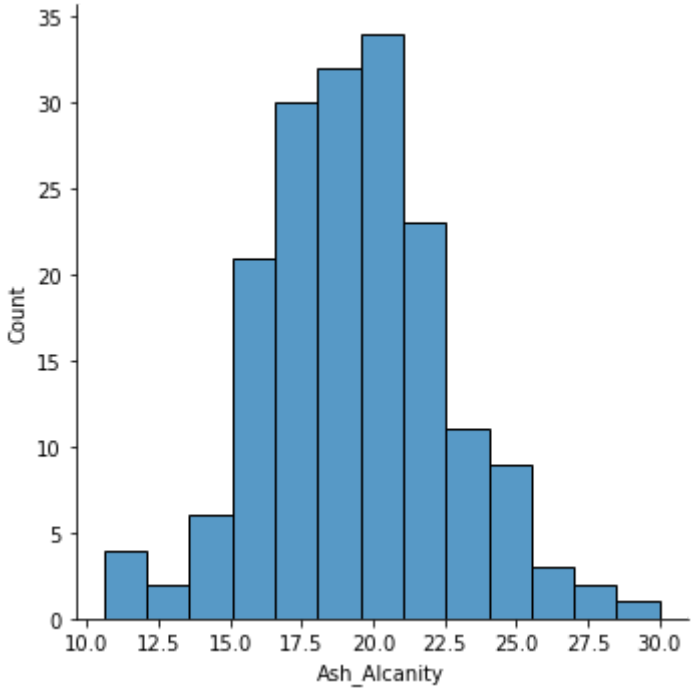
In [8]:
```
for i, col in enumerate(columns):
    plt.figure(i)
    sns.displot(wines[col])
```

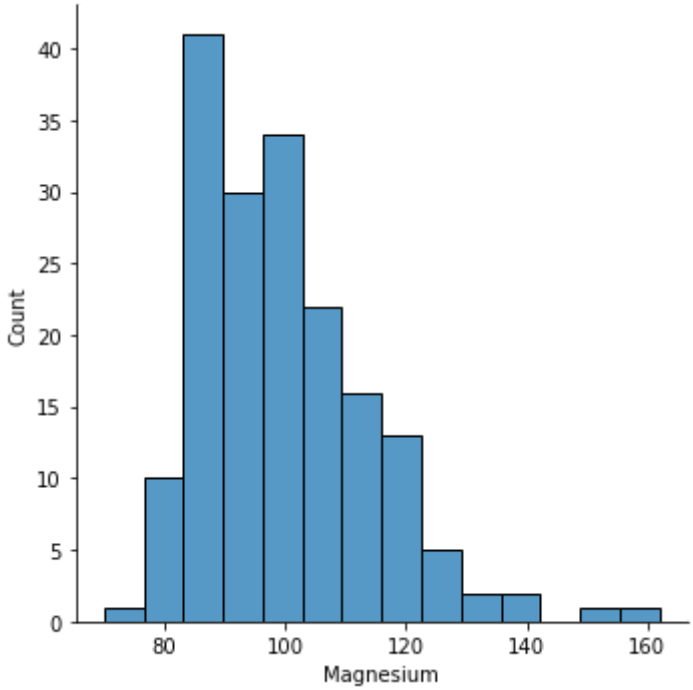
```
<Figure size 432x288 with 0 Axes>
```

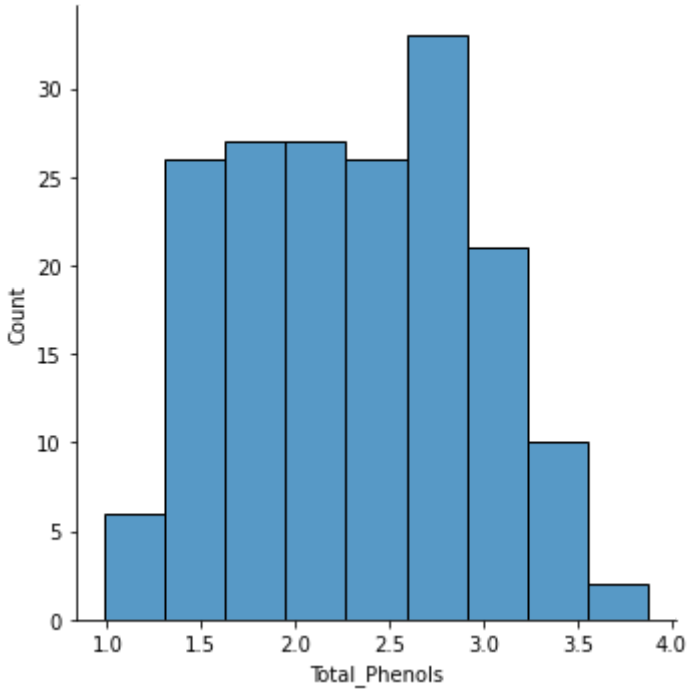

Count
Alcohol
0
5
10
15
20
25
30
11.0
11.5
12.0
12.5
13.0
13.5
14.0
14.5
15.0

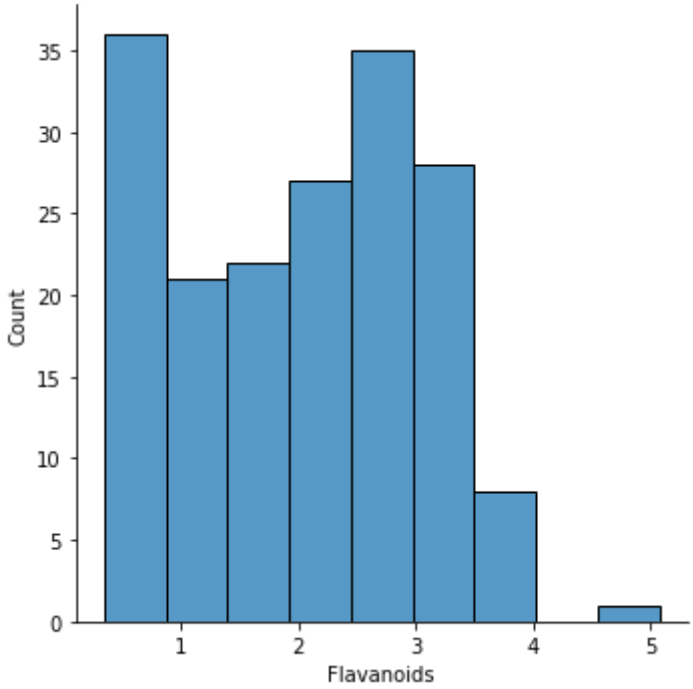

Count
Malic_Acid
0
10
20
30
40
50
60
1
2
3
4
5
6

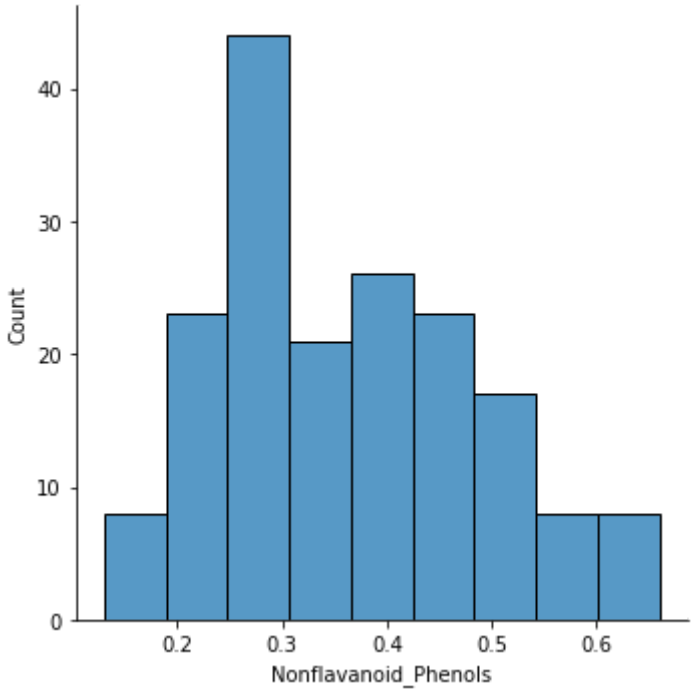

Count
Ash
0
5
10
15
20
25
30
35
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25

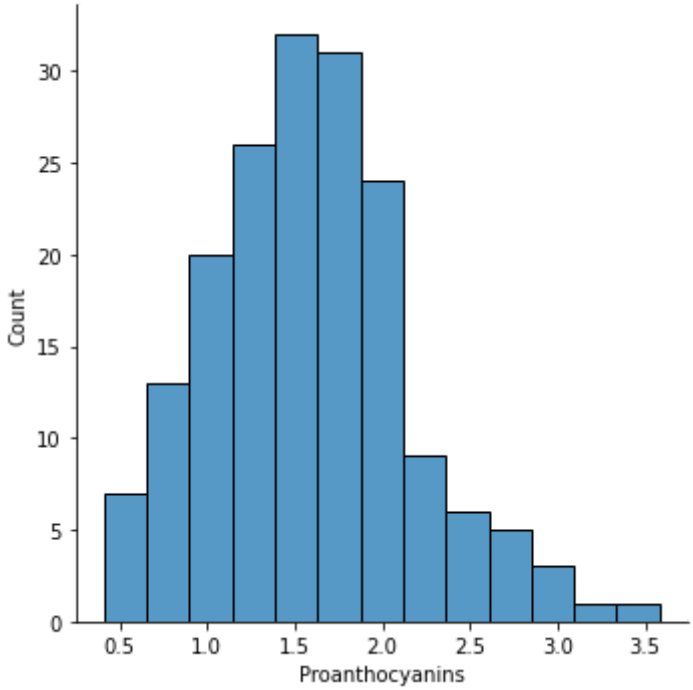

Count
Ash_Alcanity
0
5
10
15
20
25
30
35
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

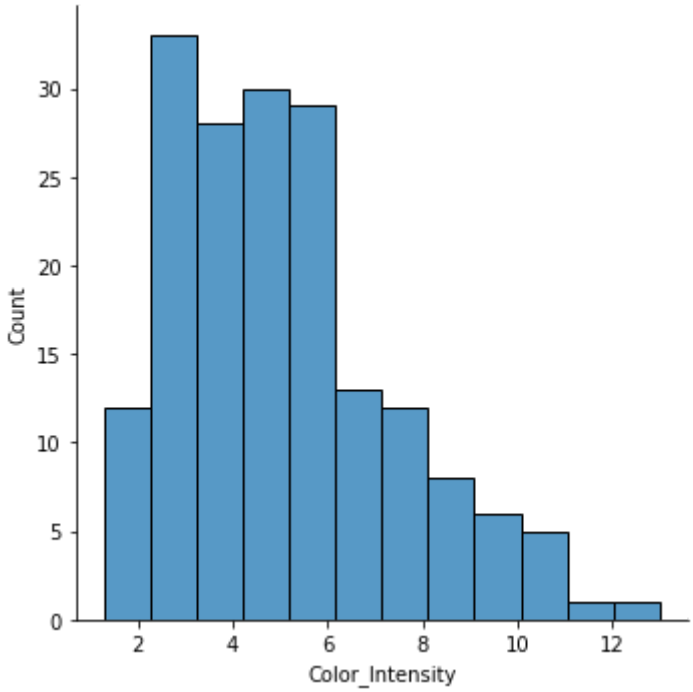

Count
Magnesium
0
5
10
15
20
25
30
35
40
80
100
120
140
160

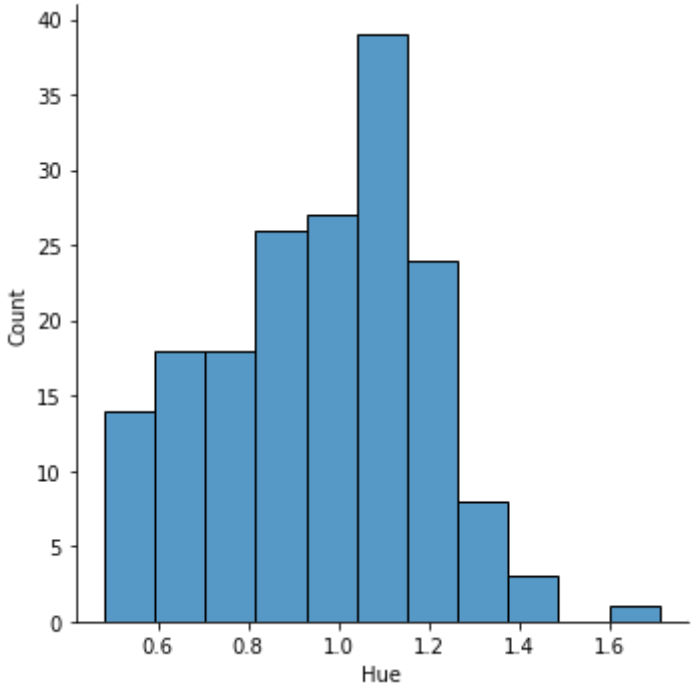

Count
Total_Phenols
0
5
10
15
20
25
30
1.0
1.5
2.0
2.5
3.0
3.5
4.0

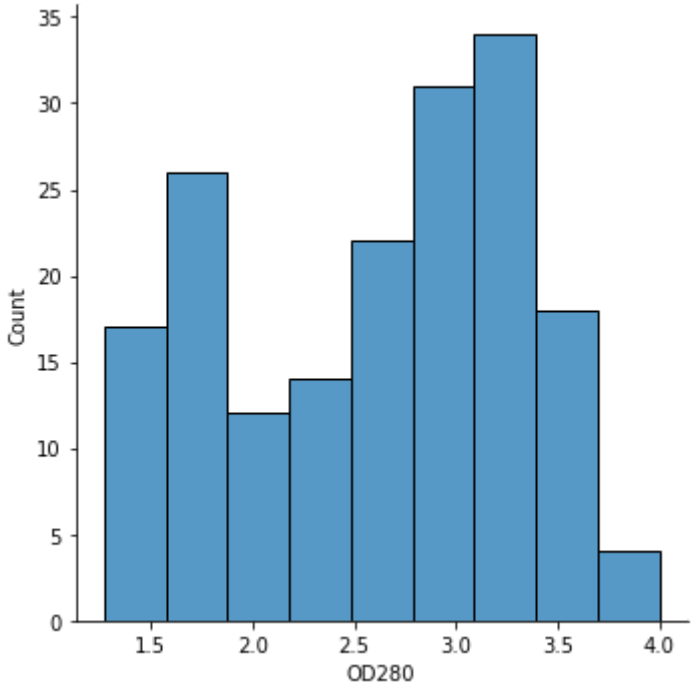

Count
Flavanoids
0
5
10
15
20
25
30
35
1
2
3
4
5

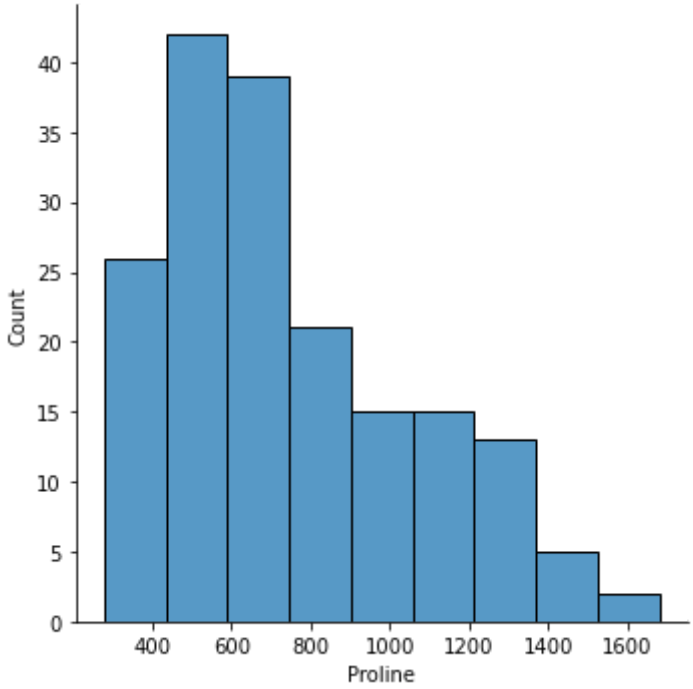

Count
Nonflavanoid_Phenols
0
10
20
30
40
0.2
0.3
0.4
0.5
0.6


Count
Proanthocyanins
0
5
10
15
20
25
30
0.5
1.0
1.5
2.0
2.5
3.0
3.5


Count
Color_Intensity
0
5
10
15
20
25
30
2
4
6
8
10
12

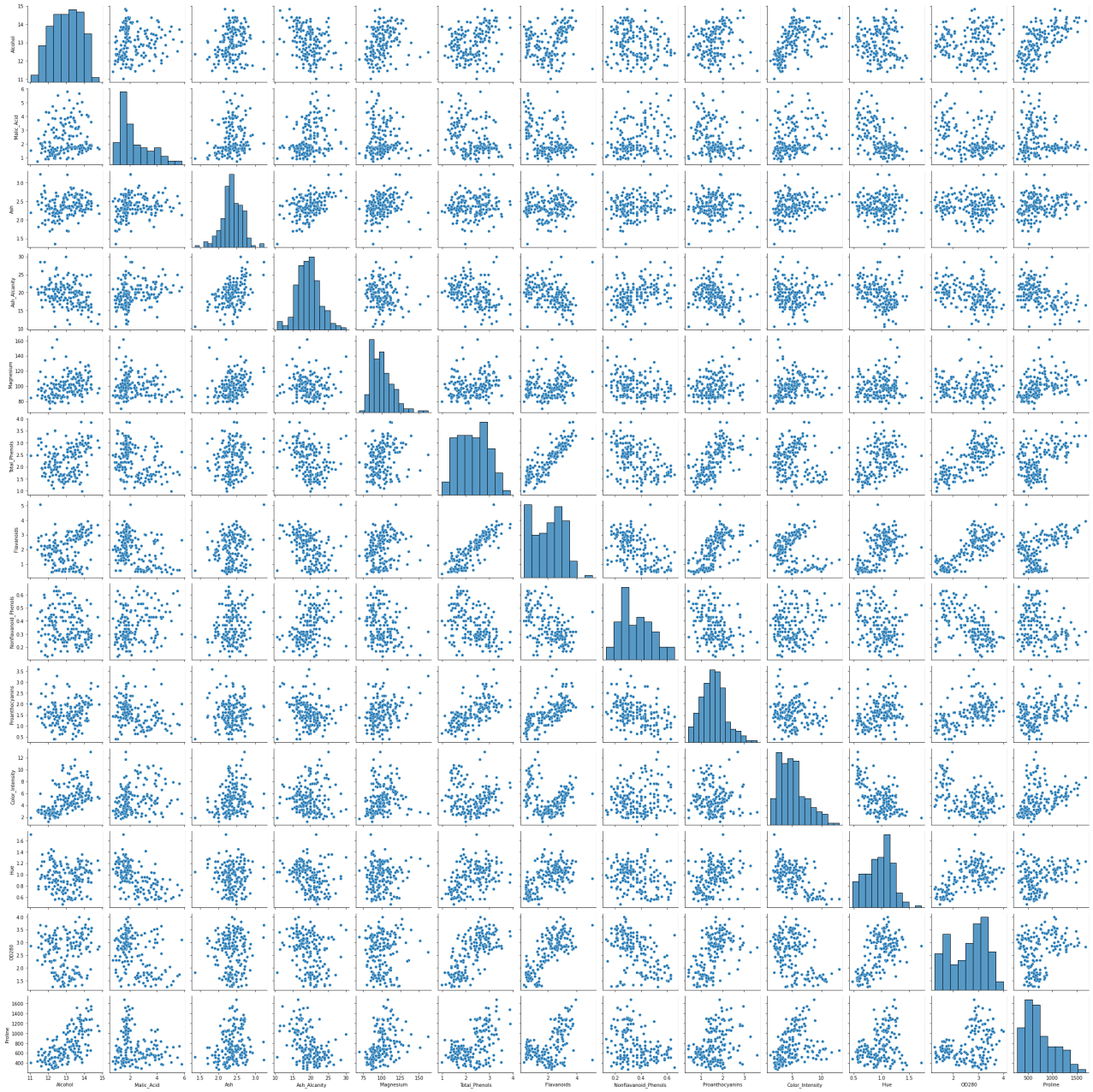

Count
Hue
0
5
10
15
20
25
30
35
40
0.6
0.8
1.0
1.2
1.4
1.6


Count
OD280
0
5
10
15
20
25
30
35
1.5
2.0
2.5
3.0
3.5
4.0



In [9]:

```
sns.pairplot(wines)
plt.show()
```



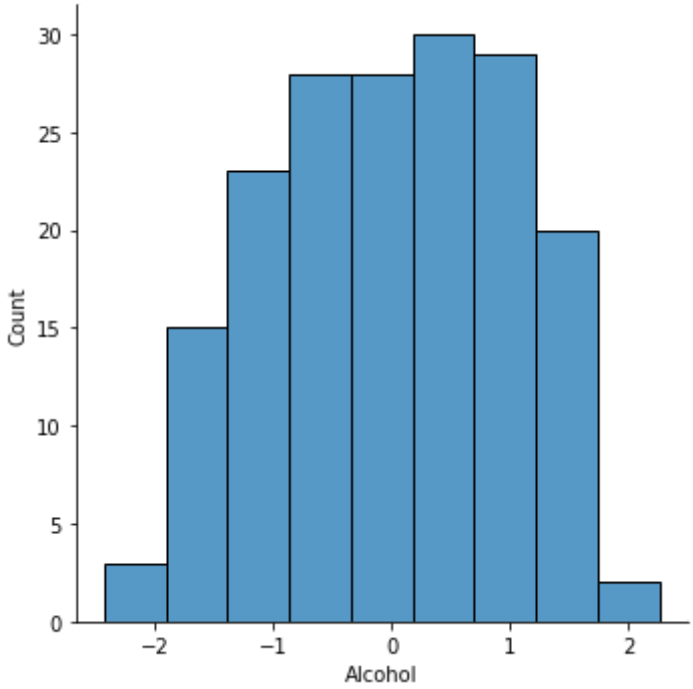
KMeans Clustering and Feature Reduction Using PCA

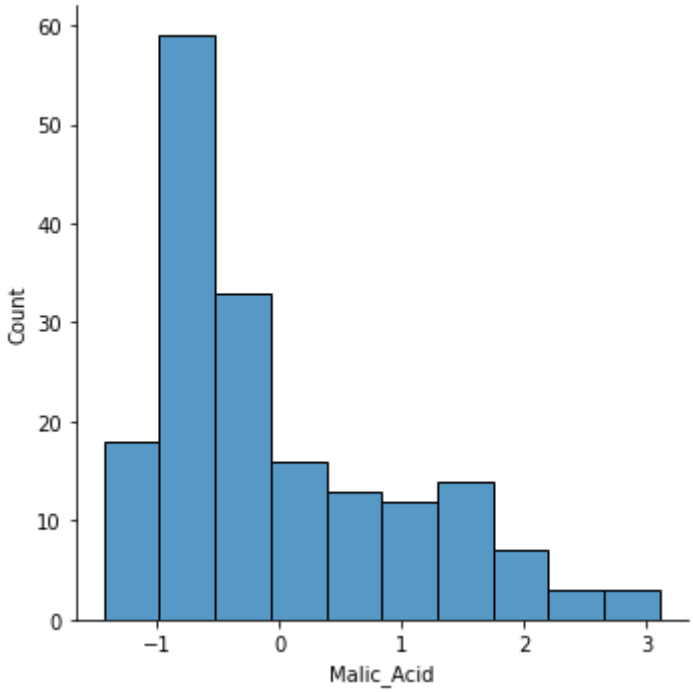
In [10]:
```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

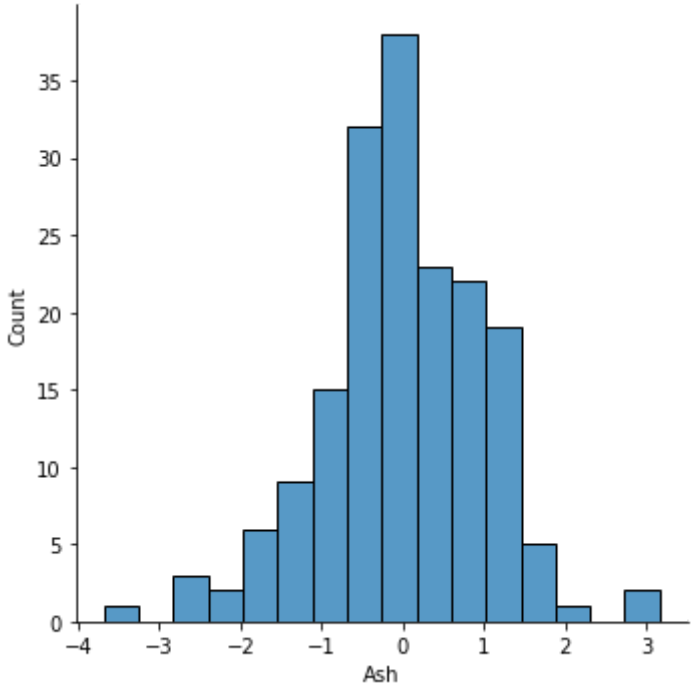
In [11]:
```
ss = StandardScaler()

scaled_df = pd.DataFrame(ss.fit_transform(wines), columns = columns)
```

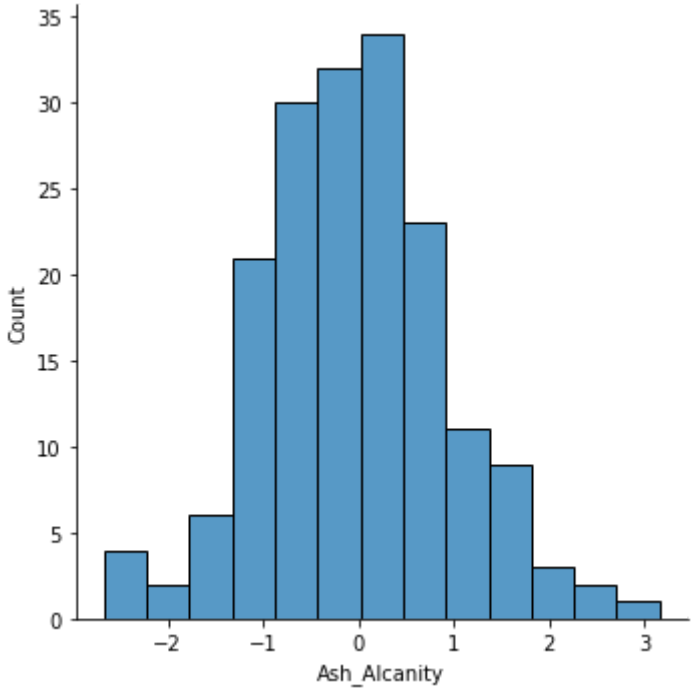
In [12]:
```
for i, col in enumerate(columns):
    plt.figure(i)
    sns.displot(scaled_df[col])
```

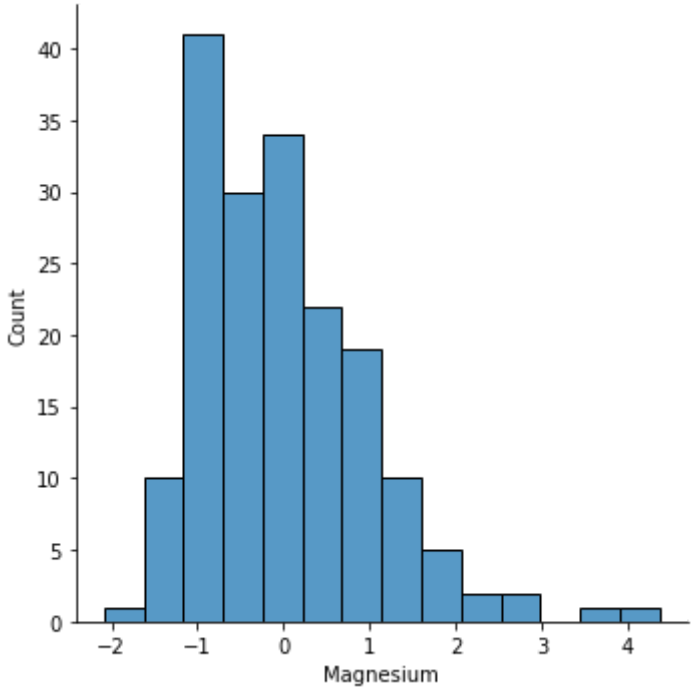
```
<Figure size 432x288 with 0 Axes>
```

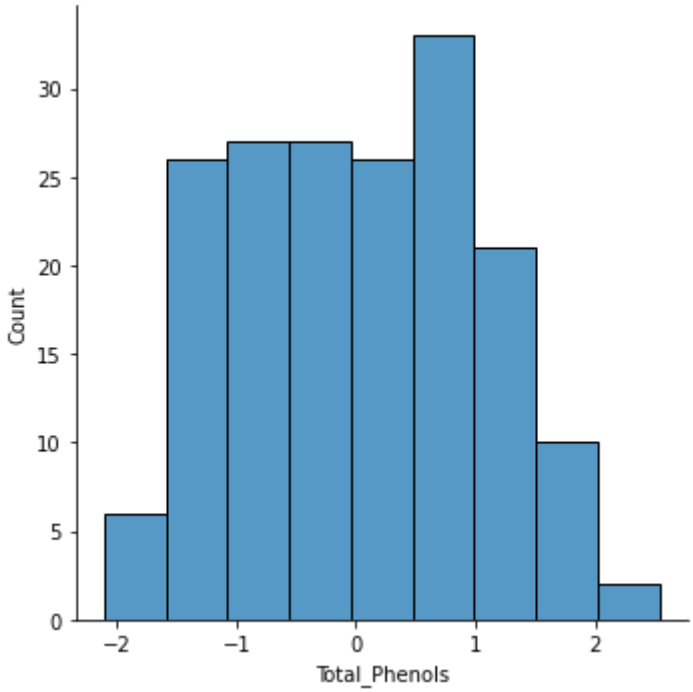

Count
Alcohol
30
25
20
15
10
5
0
−2
−1
0
1
2

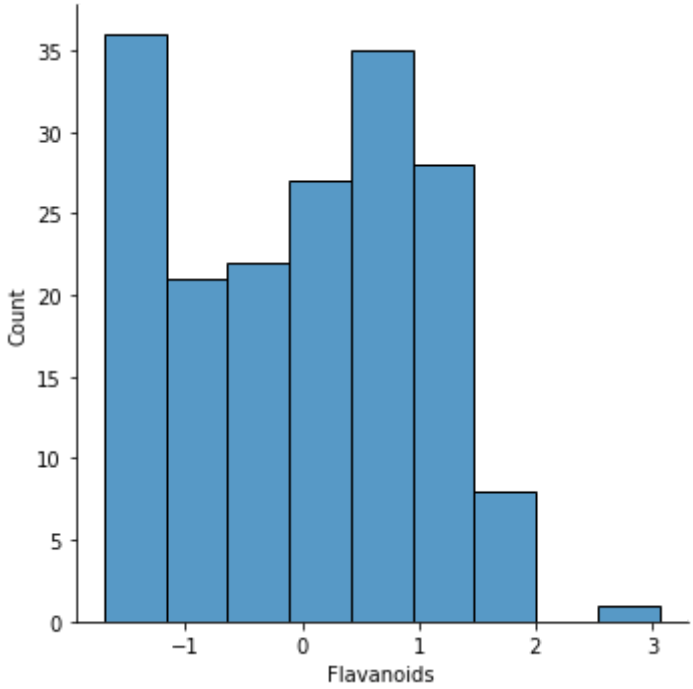

Count
Malic_Acid
60
50
40
30
20
10
0
−1
0
1
2
3

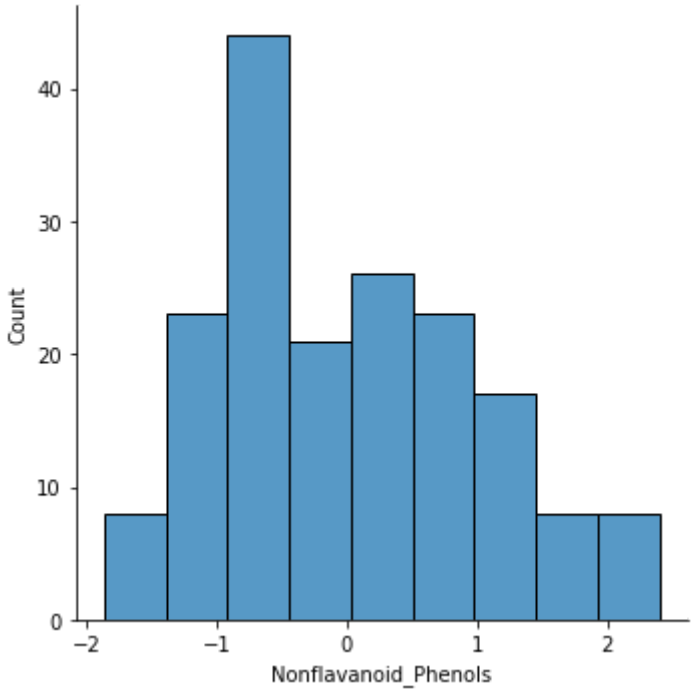

Count
Ash
35
30
25
20
15
10
5
0
−4
−3
−2
−1
0
1
2
3

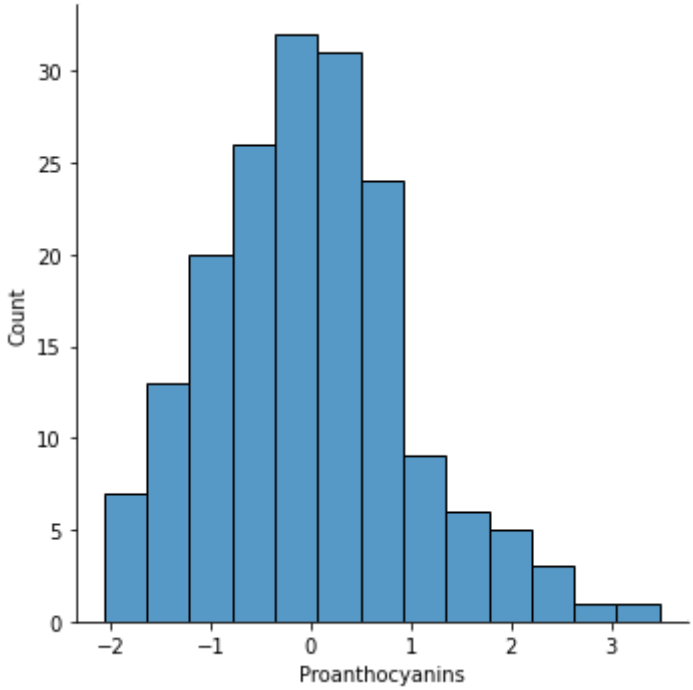

Count
Ash_Alcanity
35
30
25
20
15
10
5
0
−2
−1
0
1
2
3

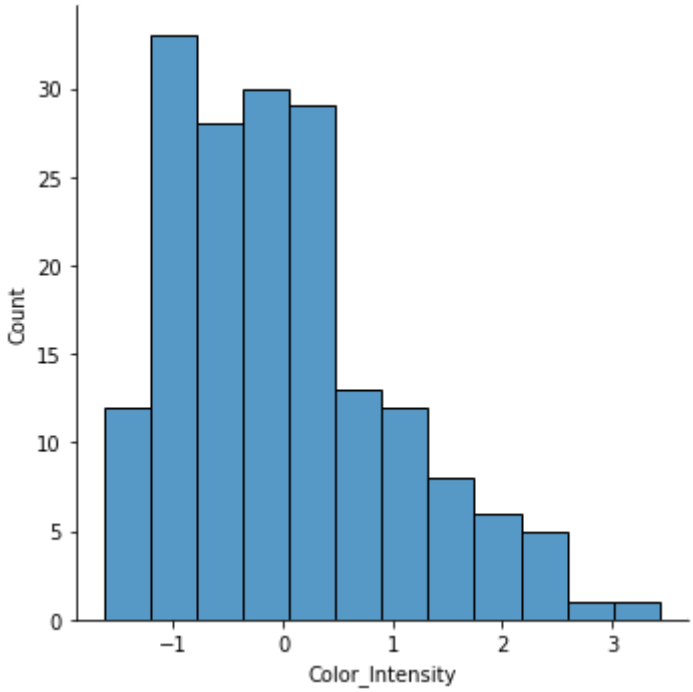

Count
Magnesium
0
5
10
15
20
25
30
35
40
−2
−1
0
1
2
3
4

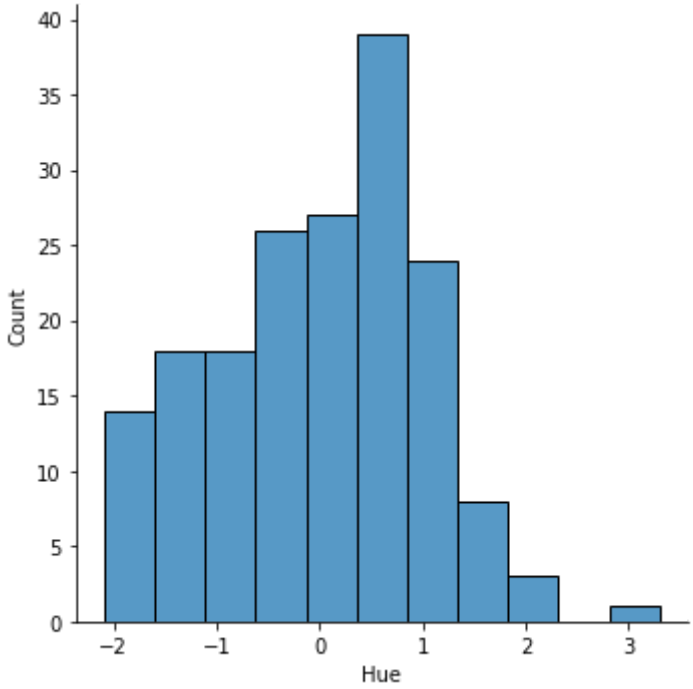

Count
Total_Phenols
0
5
10
15
20
25
30
−2
−1
0
1
2

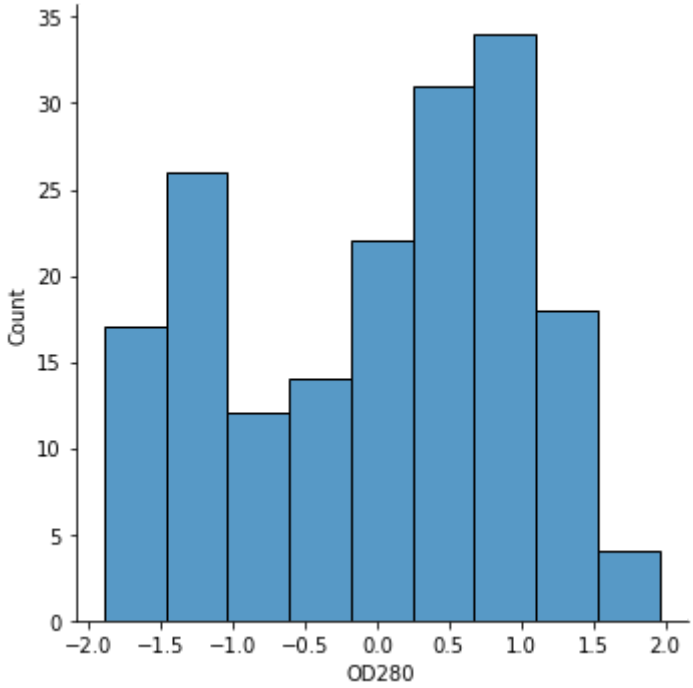

Count
Flavanoids
0
5
10
15
20
25
30
35
−1
0
1
2
3

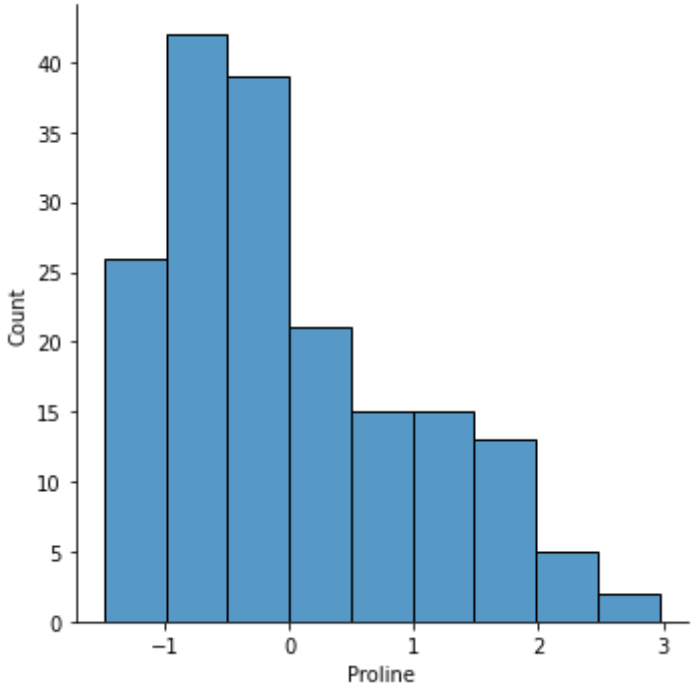

Count
Nonflavanoid_Phenols
0
10
20
30
40
−2
−1
0
1
2


Count
Proanthocyanins
0
5
10
15
20
25
30
−2
−1
0
1
2
3


Count
Color_Intensity
0
5
10
15
20
25
30
−1
0
1
2
3


Count
Hue
0
5
10
15
20
25
30
35
40
−2
−1
0
1
2
3


Count
OD280
0
5
10
15
20
25
30
35
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0



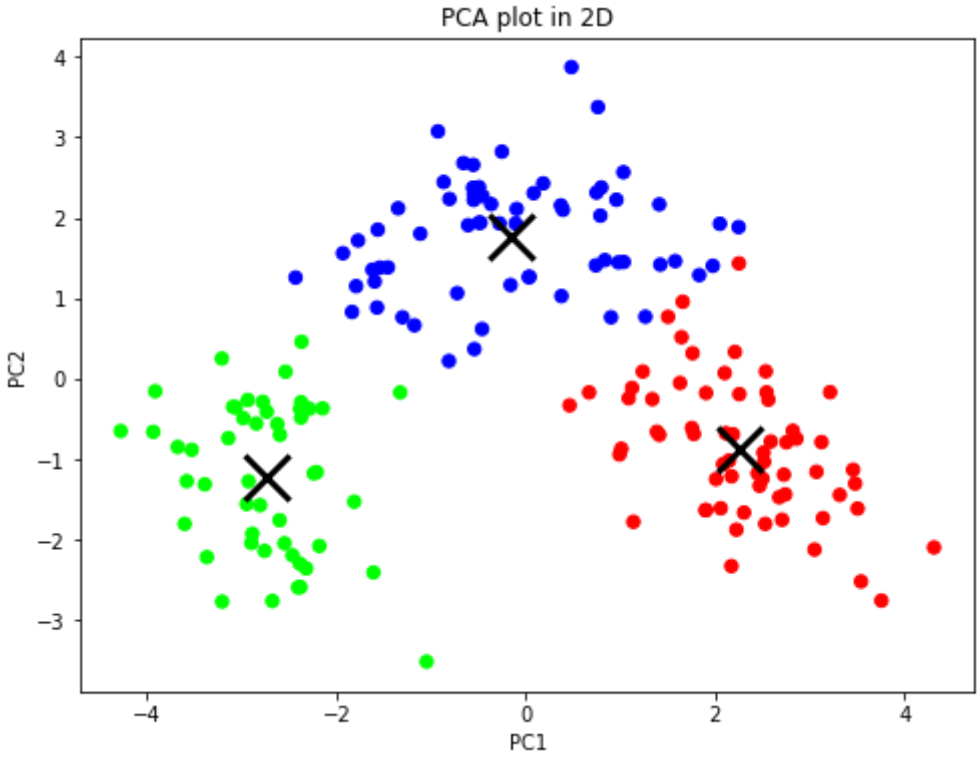
In [13]:
```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(scaled_df)

principal_df = pd.DataFrame(data = X_pca, columns = ['PCA1', 'PCA2'])
```

In [14]:
```
kmeans = KMeans(n_clusters=3, n_init=15, max_iter=500, random_state=0)
```

In [15]:
```
# Train and make predictions
kmeansclusters_pca = kmeans.fit_predict(principal_df)
```

In [16]:
```
# Cluster centers
centroids = kmeans.cluster_centers_
```

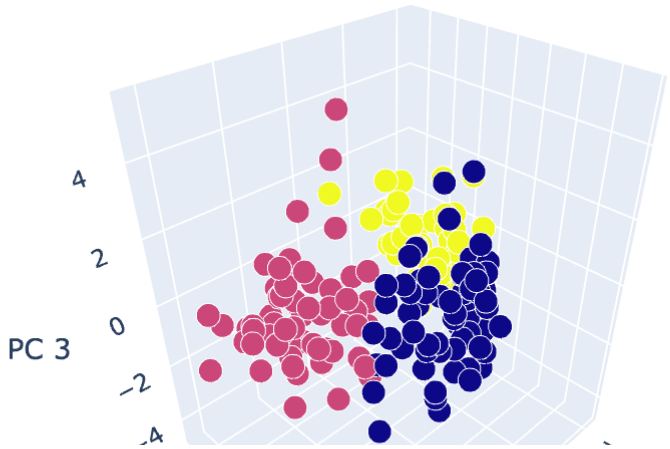
In [17]:
```
# Figure size
plt.figure(figsize=(8,6))

# Scatterplot
plt.scatter(principal_df.iloc[:,0], principal_df.iloc[:,1], c=kmeansclusters_pca
plt.scatter(x=centroids[:,0], y=centroids[:,1], marker="x", s=500, linewidths=3,

# Aesthetics
plt.title('PCA plot in 2D')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.show()
```



In [18]:

```
# PCA
pca = PCA(n_components=3)
components = pca.fit_transform(scaled_df)

# 3D scatterplot
fig = px.scatter_3d(
    components, x=0, y=1, z=2, color=kmeansclusters_pca, size=0.1*np.ones(len(sc
    title='PCA plot in 3D',
    labels={'0': 'PC 1', '1': 'PC 2', '2': 'PC 3'},
    width=800, height=500
)
fig.show()
```

PCA plot in 3D



In [19]:
```
kmeansclusters_pca
```

Out[19]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 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, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2], dtype=int32)
```

In [20]:
```
df = wines
```

In [21]:
```
df['KMeans_Cluster_PCA'] = kmeansclusters_pca.tolist()
```

In [22]:
```
df.sample(5)
```

Out[22]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavano
106	12.25	1.73	2.12	19.0	80	1.65	2.03	
143	13.62	4.95	2.35	20.0	92	2.00	0.80	
92	12.69	1.53	2.26	20.7	80	1.38	1.46	
174	13.40	3.91	2.48	23.0	102	1.80	0.75	
162	12.85	3.27	2.58	22.0	106	1.65	0.60	

KMeans Clustering and Feature Reduction Using UMAP

In [23]:
```
import umap.umap_ as umap
```

In [24]:
```
um = umap.UMAP()
x_fit = um.fit(scaled_df)
x_umap = um.transform(scaled_df)
```

In [25]:
```
umap_df = pd.DataFrame(data = x_umap, columns = ['umap comp. 1', 'umap comp. 2']
```

In [26]:
```
umap_df.shape
```

Out[26]:
```
(178, 2)
```

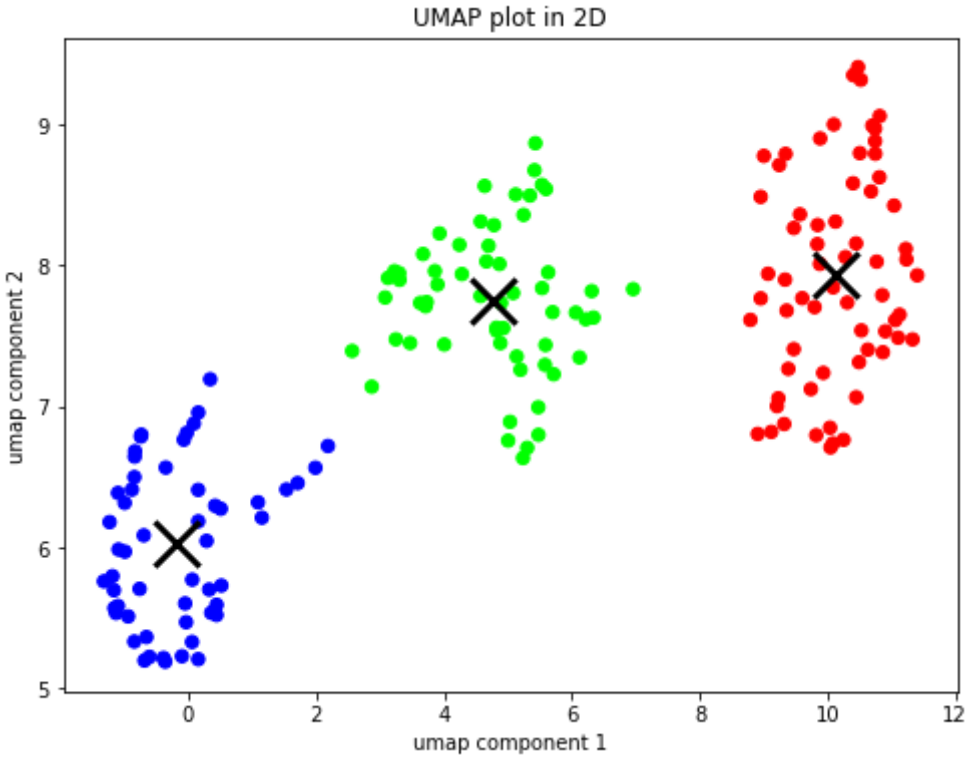
In [27]:
```
umap_df.head()
```

Out[27]:

	umap comp. 1	umap comp. 2
0	10.814951	9.059958
1	9.844379	8.287432
2	10.492686	7.315001
3	11.402284	7.930831
4	9.208191	7.003255

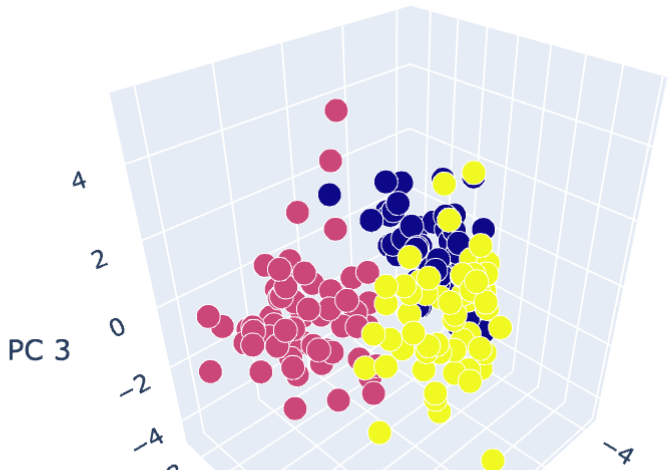
In [28]:
```
kmeans = KMeans(n_clusters=3, n_init=15, max_iter=500, random_state=0)

# Train and make predictions
kmeansclusters_umap = kmeans.fit_predict(x_umap)

# Cluster centers
centroids = kmeans.cluster_centers_
```

In [29]:
```
plt.figure(figsize=(8,6))

# Scatterplot
plt.scatter(umap_df.iloc[:,0], umap_df.iloc[:,1], c=kmeansclusters_umap, cmap="b

# Centroids
plt.scatter(x=centroids[:,0], y=centroids[:,1], marker="x", s=500, linewidths=3,

# Aesthetics
plt.title('UMAP plot in 2D')
plt.xlabel('umap component 1')
plt.ylabel('umap component 2')
plt.show()
```



In [30]:
```
# PCA
um = umap.UMAP(n_components=3)
components_umap = um.fit_transform(scaled_df)

# 3D scatterplot
fig = px.scatter_3d(
    components, x=0, y=1, z=2, color=kmeansclusters_umap, size=0.1*np.ones(len(s
    title='UMAP plot in 3D',
    labels={'0': 'PC 1', '1': 'PC 2', '2': 'PC 3'},
    width=800, height=500
)
fig.show()
```

UMAP plot in 3D



In [31]:
```
kmeansclusters_umap
```

Out[31]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 2, 2, 2, 2,
       2, 2, 0, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 0, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0], dtype=int32)
```

In [32]:
```
df['KMeans_Cluster_UMAP'] = kmeansclusters_umap.tolist()
```

In [33]:
```
df.sample(5)
```

Out[33]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoi
30	13.73	1.50	2.70	22.5	101	3.00	3.25	
167	12.82	3.37	2.30	19.5	88	1.48	0.66	
63	12.37	1.13	2.16	19.0	87	3.50	3.10	
13	14.75	1.73	2.39	11.4	91	3.10	3.69	
177	14.13	4.10	2.74	24.5	96	2.05	0.76	

KMeans Clustering and Feature Reduction Using t-SNE

In [34]:
```
from sklearn.manifold import TSNE
```

In [35]:
```
tsne = TSNE(n_components=2)
x_tsne = tsne.fit_transform(scaled_df)
```

In [36]:
```
tsne_df = pd.DataFrame(data = x_tsne, columns = ['tsne comp. 1', 'tsne comp. 2']
```

In [37]:
```
tsne_df.shape
```

Out[37]:
```
(178, 2)
```

In [38]:
```
tsne_df.head()
```

Out[38]:

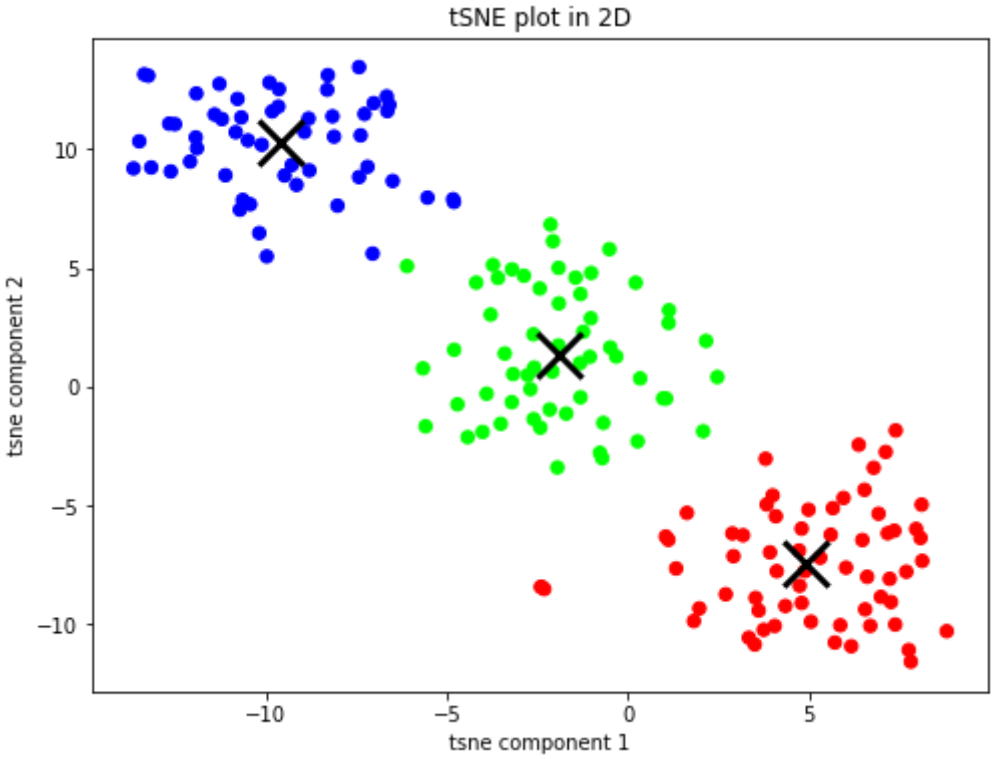
	tsne comp. 1	tsne comp. 2
0	3.507671	-10.853365
1	2.917411	-7.133722
2	7.697696	-7.788281
3	6.175278	-10.926777
4	6.795211	-3.410044

In [39]:

```
kmeans = KMeans(n_clusters=3, n_init=15, max_iter=500, random_state=0)

# Train and make predictions
kmeansclusters_tsne = kmeans.fit_predict(x_tsne)

# Cluster centers
centroids = kmeans.cluster_centers_
```

In [40]:

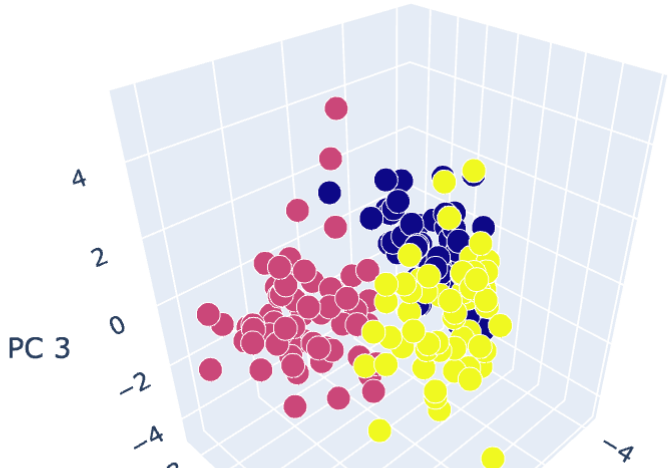
```
plt.figure(figsize=(8,6))

# Scatterplot
plt.scatter(tsne_df.iloc[:,0], tsne_df.iloc[:,1], c=kmeansclusters_tsne, cmap="b

# Centroids
plt.scatter(x=centroids[:,0], y=centroids[:,1], marker="x", s=500, linewidths=3,

# Aesthetics
plt.title('tSNE plot in 2D')
plt.xlabel('tsne component 1')
plt.ylabel('tsne component 2')
plt.show()
```



In [41]:

```
# tSNE
tsne = TSNE(n_components=3)
components_tsne = tsne.fit_transform(scaled_df)

# 3D scatterplot
fig = px.scatter_3d(
    components, x=0, y=1, z=2, color=kmeansclusters_tsne, size=0.1*np.ones(len(s
    title='tSNE plot in 3D',
    labels={'0': 'PC 1', '1': 'PC 2', '2': 'PC 3'},
    width=800, height=500
)
fig.show()
```

tSNE plot in 3D



In [42]:
```
kmeansclusters_tsne
```

Out[42]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 2, 2, 2, 2,
       2, 2, 0, 1, 0, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 0, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0], dtype=int32)
```

In [43]:
```
df['KMeans_Cluster_tSNE'] = kmeansclusters_tsne.tolist()
```

In [44]:
```
df.sample(5)
```

Out[44]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavano
25	13.05	2.05	3.22	25.0	124	2.63	2.68	
150	13.50	3.12	2.62	24.0	123	1.40	1.57	
28	13.87	1.90	2.80	19.4	107	2.95	2.97	
166	13.45	3.70	2.60	23.0	111	1.70	0.92	
157	12.45	3.03	2.64	27.0	97	1.90	0.58	

In [45]:
```
kmeansclusters_tsne
```

Out[45]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 2, 2, 2, 2,
       2, 2, 0, 1, 0, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 0, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0], dtype=int32)
```

In [46]:
```
kmeansclusters_umap
```

Out[46]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 2, 0, 0, 2, 2, 2, 2,
       2, 2, 0, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 0, 2, 2, 2, 2,
```

```
       2, 2, 2, 2, 2, 2, 2, 1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 1, 2, 2, 2, 2, 2, 2, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0], dtype=int32)
```

In [47]:
```
kmeansclusters_pca
```

Out[47]:
```
array([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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 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, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2], dtype=int32)
```

In [50]:
```
df['KMeans_Cluster_PCA'] = df.KMeans_Cluster_PCA.replace({0: 2, 2: 0})
```

In [54]:
```
df.sample(10)
```

Out[54]:

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavano
42	13.88	1.89	2.59	15.0	101	3.25	3.56	
169	13.40	4.60	2.86	25.0	112	1.98	0.96	
126	12.43	1.53	2.29	21.5	86	2.74	3.15	
164	13.78	2.76	2.30	22.0	90	1.35	0.68	
73	12.99	1.67	2.60	30.0	139	3.30	2.89	
146	13.88	5.04	2.23	20.0	80	0.98	0.34	
98	12.37	1.07	2.10	18.5	88	3.52	3.75	
22	13.71	1.86	2.36	16.6	101	2.61	2.88	
93	12.29	2.83	2.22	18.0	88	2.45	2.25	
6	14.39	1.87	2.45	14.6	96	2.50	2.52	

Export to CSV

Exporting to .csv file in order to start using the clusters in Tableau Data Visualization

In [52]:
```
df.to_csv('wines_with_clusters.csv')
```

In [56]:
```
df1 = df[['KMeans_Cluster_PCA', 'KMeans_Cluster_UMAP', 'KMeans_Cluster_tSNE']]
```

In [62]:
```
df1.KMeans_Cluster_PCA.value_counts()
```

Out[62]:
```
2    65
1    64
```

```
0    49
Name: KMeans_Cluster_PCA, dtype: int64
```

In [63]:
```
df1.KMeans_Cluster_UMAP.value_counts()
```

Out[63]:
```
1    65
2    59
0    54
Name: KMeans_Cluster_UMAP, dtype: int64
```

In [64]:
```
df1.KMeans_Cluster_tSNE.value_counts()
```

Out[64]:
```
1    65
2    58
0    55
Name: KMeans_Cluster_tSNE, dtype: int64
```

In [ ]:
```

```