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

```
ds = pd.read_csv("wine-clustering.csv")
```

```
ds.head()
```

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

```
ds.shape
```

```
(178, 13)
```

```
ds.info()
```

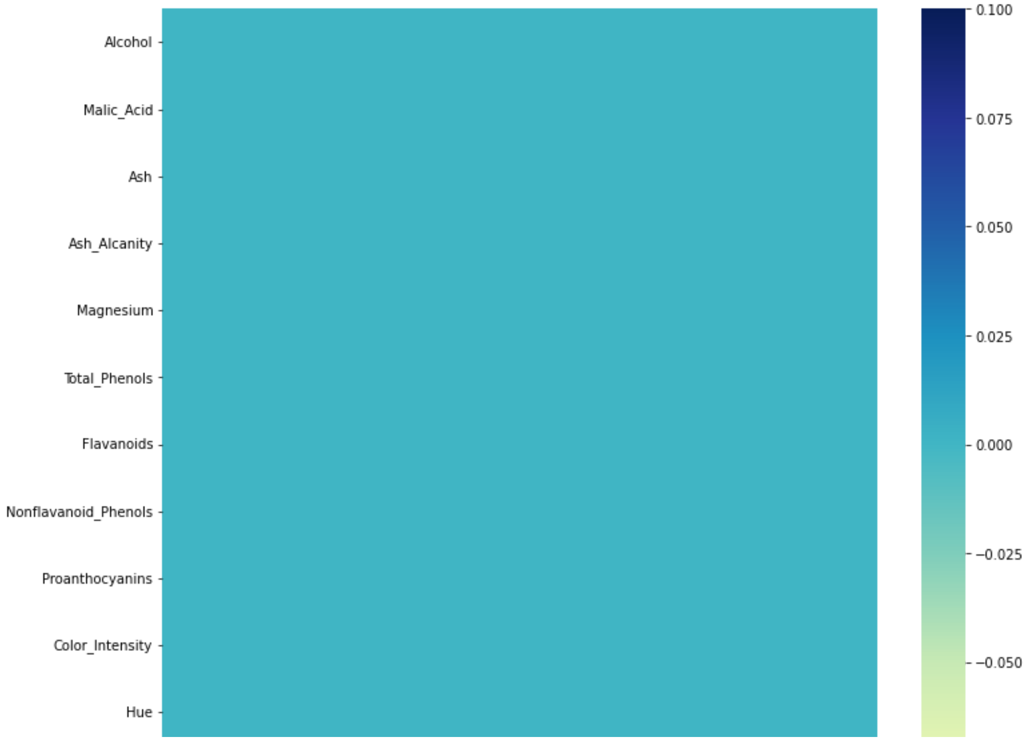
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Alcohol               178 non-null    float64
 1   Malic_Acid            178 non-null    float64
 2   Ash                   178 non-null    float64
 3   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
```

```
ds.isnull().sum()
```

```
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
```

```
plt.figure(figsize=(12,12))
sns.heatmap(ds.isna().transpose(),cmap='YlGnBu')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f0e13fb50>
```

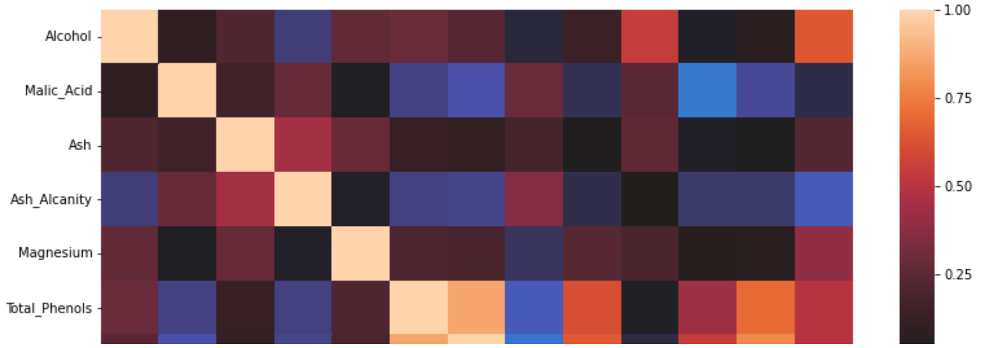


```
ds.describe()
```

	Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflava
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	

```
plt.figure(figsize=(13,10))
corr = ds.corr()
sns.heatmap(corr, vmin=-1, center=0, vmax=1)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3f0b01e370>
```



```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
ds = sc.fit_transform(ds)
```

```
ds
```

```
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)
ds = pca.fit_transform(ds)
```

```
ds
```

```
       [-1.0481819 , -3.51508969],
       [-1.60991228, -2.40663816],
       [-3.14313097, -0.73816104],
       [-2.2401569 , -1.17546529],
       [-2.84767378, -0.55604397],
       [-2.59749706, -0.69796554],
       [-2.94929937, -1.55530896],
       [-3.53003227, -0.8825268 ],
       [-2.40611054, -2.59235618],
       [-2.92908473, -1.27444695],
       [-2.18141278, -2.07753731],
       [-2.38092779, -2.58866743],
       [-3.21161722,  0.2512491 ],
       [-3.67791872, -0.84774784],
       [-2.4655558 , -2.1937983 ],
       [-3.37052415, -2.21628914],
       [-2.60195585, -1.75722935],
       [-2.67783946, -2.76089913],
       [-2.38701709, -2.29734668],
       [-3.20875816, -2.76891957]])
```

```
ds = pd.DataFrame(columns=['x','y'], data=ds)
ds
```

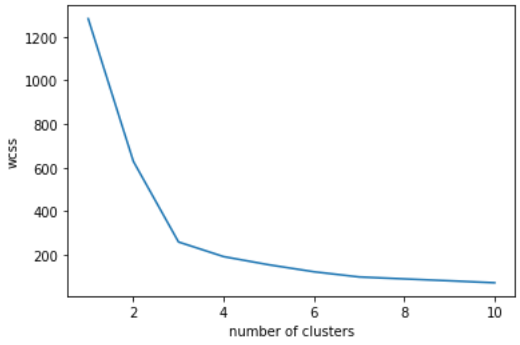
	x	y
0	3.316751	-1.443463
1	2.209465	0.333393
2	2.516740	-1.031151
3	3.757066	-2.756372
4	1.008908	-0.869831
...	...	...
173	-3.370524	-2.216289
174	-2.601956	-1.757229
175	-2.677839	-2.760899
176	-2.387017	-2.297347
177	-3.208758	-2.768920

178 rows × 2 columns

```
ds.shape
```

```
(178, 2)
```

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
  kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
  kmeans.fit(ds)
  wcss.append(kmeans.inertia_)
plt.plot(range(1,11), wcss)
plt.xlabel("number of clusters")
plt.ylabel("wcss")
plt.show()
```



```
kmeans = KMeans(n_clusters=3, init='k-means++', random_state=42)
kmeans.fit(ds)
```

```
KMeans(n_clusters=3, random_state=42)
```

```
# ykmeans = kmeans.predict(ds)
ds['Labels'] = kmeans.labels_
```

```
ds
```

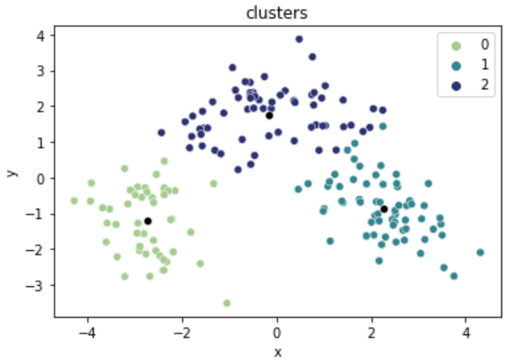
	x	y	Labels
0	3.316751	-1.443463	1
1	2.209465	0.333393	1
2	2.516740	-1.031151	1
3	3.757066	-2.756372	1
4	1.008908	-0.869831	1
...	...	...	...
173	-3.370524	-2.216289	0
174	-2.601956	-1.757229	0
175	-2.677839	-2.760899	0
176	-2.387017	-2.297347	0
177	-3.208758	-2.768920	0

178 rows × 3 columns

```
ds['Labels'].values
```

```
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, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 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)
```

```
centroids = kmeans.cluster_centers_
cen_x = centroids[:,0]
cen_y = centroids[:,1]
sns.scatterplot(data = ds, x = ds['x'], y = ds['y'], hue = ds['Labels'], palette = 'crest')
sns.scatterplot(x = cen_x, y = cen_y, c = ['black'])
plt.title("clusters")
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```


clusters
y
x