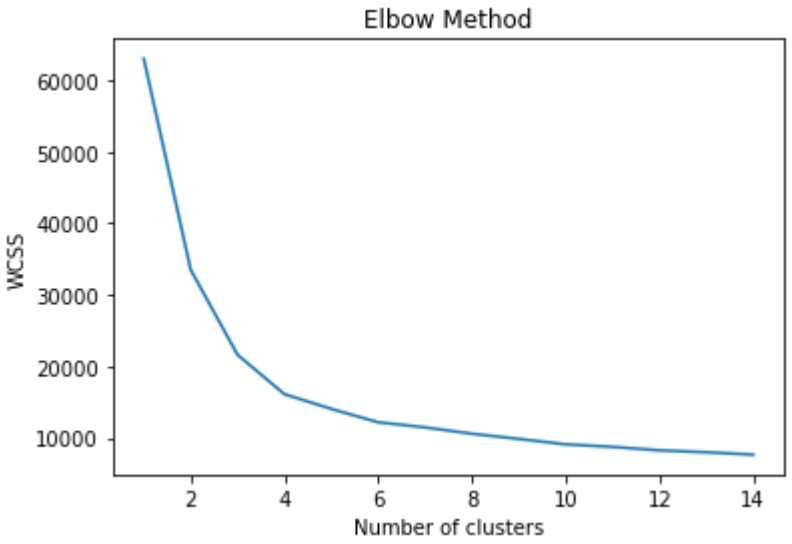
In [9]:

```
import csv
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split

#I've converted the .data file to .csv which is read and replaced ? to NaN in the files
COLUMNS_COUNT = 2

with open('water-treatment.data', 'r') as f:
    columns = [next(f).strip() for line in range(COLUMNS_COUNT)]
temp_df = pd.read_csv('water-treatment.data', skiprows=COLUMNS_COUNT, header=None, delimit
er=';', skip_blank_lines=True)
even_df = temp_df.iloc[::2].reset_index(drop=True)
odd_df = temp_df.iloc[1::2].reset_index(drop=True)
df = pd.concat([even_df, odd_df], axis=1)
df.columns = columns
df.to_csv('out.csv', index=False)
text = open("out.csv", "r")
text = ''.join([i for i in text]) \
    .replace("?", "NaN")
x = open("out.csv","w")
x.writelines(text)
x.close()

reader=pd.read_csv('water-treatment.csv',header=None,delimiter=',');
df=pd.DataFrame(reader)
# print('Before Cleaning Up the DataSet\n')
# print(df)

#Calculating the Median of each Column and Replacing "NaN" with the Corresponding Median v
alues
for i in range(1,39):
    mean = df.loc[:,i].mean()
    # print('The mean of column :'+str(i))
    # print(mean)
    df.loc[:,i].fillna(mean, inplace=True)

for i in range(1,39):
    for j in range(0,527):
        mean=df.loc[:,i].mean();
        stdevi=df.loc[:,i].std();
        df.loc[j,i]=(df.loc[j,i]-mean)/stdevi;

print('\n')
# print('After Cleaning Up the DataSet and performing Normalization\n')
# print(df)

#Dropping the Date Column
print('\n')
# print('After Dropping\n')
df.drop(df.columns[0], axis=1, inplace=True)
```

```
# print(df)

#Running the K-Means Clustering algorithm multiple times to find the Elbow Point for Effec
tive K Value
wcss = []
for i in range(1, 15):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=
0)
    kmeans.fit(df)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 15), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [ ]: