Chapter 11: Demo K-Means

In [1]:
```
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn import metrics
from scipy.spatial.distance import cdist
import pandas as pd
```

In [2]:
```
X = pd.read_excel("example.xlsx", index_col=0)
```

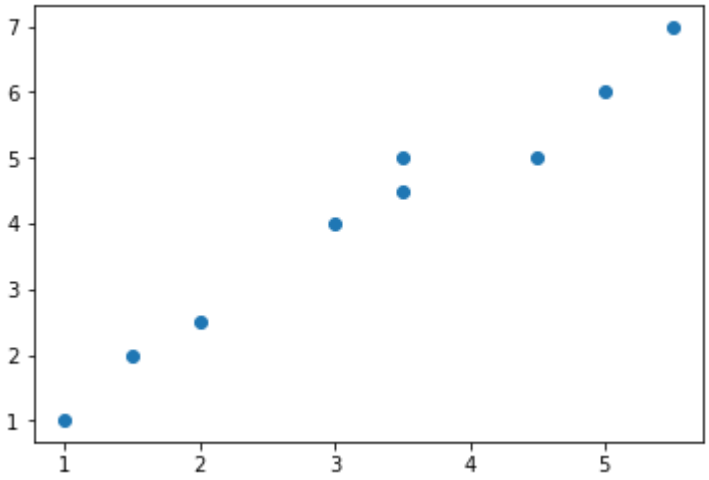
In [3]:
```
X.head(3)
```

Out[3]:

Subject	A	B	C
1	1.0	1.0	2.0
2	1.5	2.0	3.0
3	2.0	2.5	2.0

In [4]:
```
plt.scatter(X.A, X.B)
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x25bbc9ca7b8>
```



Clustering with 2 attributes

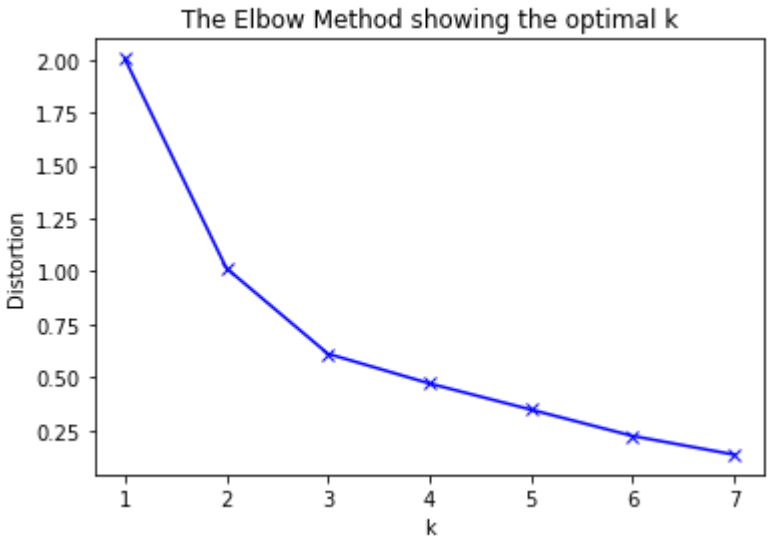
In [5]:
```
X2 = X[['A','B']]
X2.head(3)
```

Out[5]:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	2.0	2.5

In [6]:
```
# k means determine k
distortions = []
K = range(1,8)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(X2)
    kmeanModel.fit(X2)
    distortions.append(sum(np.min(cdist(X2, kmeanModel.cluster_centers_,
                                        'euclidean'), axis=1)) / X.shape[0])
```

In [7]:
```
# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



In [8]:
```
# k = 3
kmeans = KMeans(n_clusters=3)
kmeans.fit(X2)
```

Out[8]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [9]:
```
centroids = kmeans.cluster_centers_
labels = kmeans.labels_
print(centroids)
print(labels)
```

```
[[1.5        1.83333333]
 [3.625      4.625     ]
 [5.25       6.5       ]]
[0 0 0 1 2 1 1 1 2]
```

In [10]:
```
X2['Group'] = pd.Series(labels)
X2.head()
```

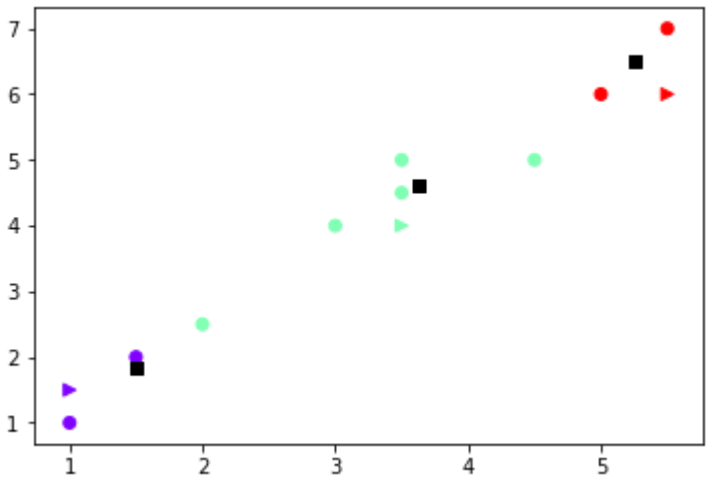
Out[10]:

Subject	A	B	Group
1	1.0	1.0	0
2	1.5	2.0	0
3	2.0	2.5	1
3	3.0	4.0	1
4	5.0	6.0	2

In [11]:
```
X_test = np.array([[1, 1.5], [3.5,4], [5.5,6]])
pred = kmeans.predict(X_test)
pred
```

Out[11]:
```
array([0, 1, 2])
```

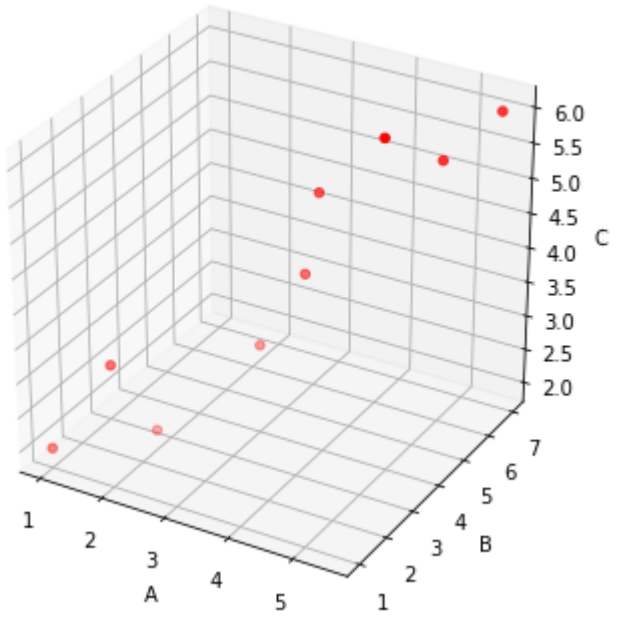
In [12]:
```
plt.scatter(X2.A, X2.B, c= X2.Group, cmap='rainbow')
plt.scatter(centroids[:, 0],centroids[:, 1], marker = "s",c='black')
plt.scatter(X_test[:,0],X_test[:,1],
            marker=">", c=pred, cmap='rainbow')
plt.show()
```



Clustering with 3 attributes

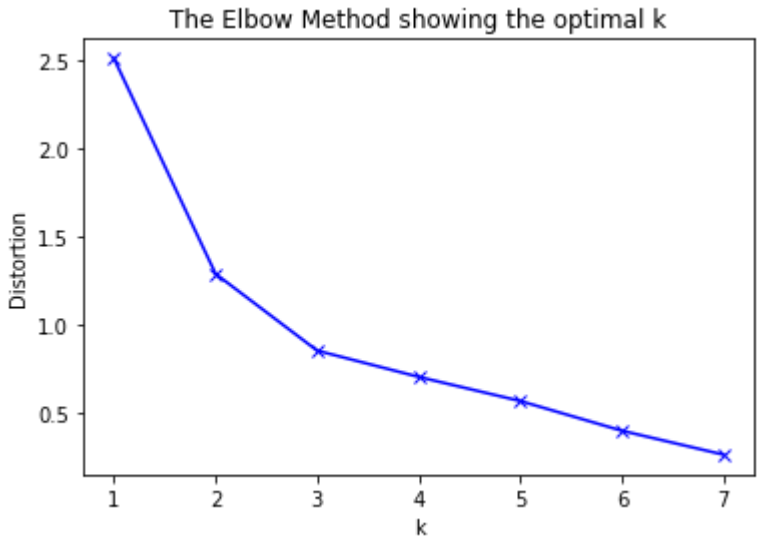
In [13]:
```
from mpl_toolkits.mplot3d import Axes3D
```

In [14]:
```
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X.A, X.B, X.C, c='r', marker='o')
ax.set_xlabel('A')
ax.set_ylabel('B')
ax.set_zlabel('C')
plt.show()
```



In [15]:
```
# k means determine k
distortions = []
K = range(1,8)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(X)
    kmeanModel.fit(X)
    distortions.append(sum(np.min(cdist(X, kmeanModel.cluster_centers_,
                                        'euclidean'), axis=1)) / X.shape[0])
```

In [16]:
```
# Plot the elbow
plt.plot(K, distortions, 'bx-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



In [17]:
```
# k = 3
kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
```

Out[17]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [18]:
```
centroids = kmeans.cluster_centers_
labels = kmeans.labels_

print(centroids)
print(labels)
```

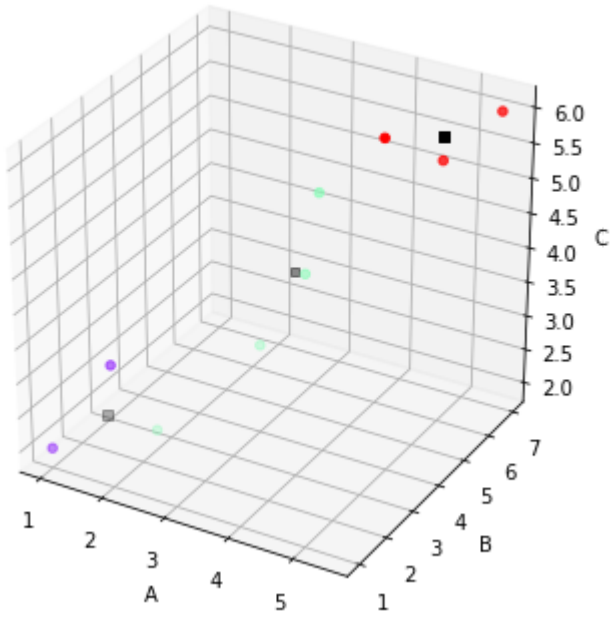
```
[[1.5        1.83333333 2.33333333]
 [3.33333333 4.5        4.        ]
 [5.         6.         5.83333333]]
[0 0 0 1 2 1 2 1 2]
```

In [19]:
```
X['Group'] = pd.Series(labels)
X.head()
```

Out[19]:

Subject	A	B	C	Group
1	1.0	1.0	2.0	0
2	1.5	2.0	3.0	0
3	2.0	2.5	2.0	1
3	3.0	4.0	3.0	1
4	5.0	6.0	5.5	2

In [20]:
```
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X.A, X.B, X.C, c=X.Group, marker='o', cmap='rainbow')
ax.scatter(centroids[:, 0],centroids[:, 1], centroids[:, 2],
           marker = 's', c='black')
ax.set_xlabel('A')
ax.set_ylabel('B')
ax.set_zlabel('C')
plt.show()
```



In [ ]: