Chapter 11 - Exercise 1: Find Group

Cho dữ liệu data.csv. Hãy thực hiện bài toán phân cụm cho dữ liệu.

1. Đọc dữ liệu, chuẩn hóa dữ liệu (nếu cần)
2. Trực quan hóa dữ liệu
3. Áp dụng Elbow tìm k
4. Áp dụng thuật toán K-Means để giải bài toán phân cụm theo K
5. Trực quan hóa kết quả, nhận xét

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
```

In [2]:
```
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter11_Kmeans/'
```

In [3]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn import metrics
from scipy.spatial.distance import cdist
```

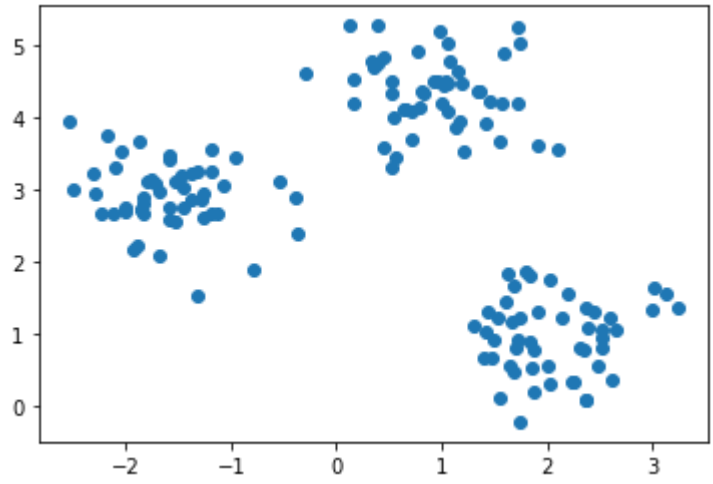
In [4]:
```
df = pd.read_csv("data.csv", index_col=0)
df.head(3)
```

Out[4]:

	f1	f2
0	2.605097	1.225296
1	0.532377	3.313389
2	0.802314	4.381962

In [5]:
```
plt.scatter(df.f1,df.f2)
```

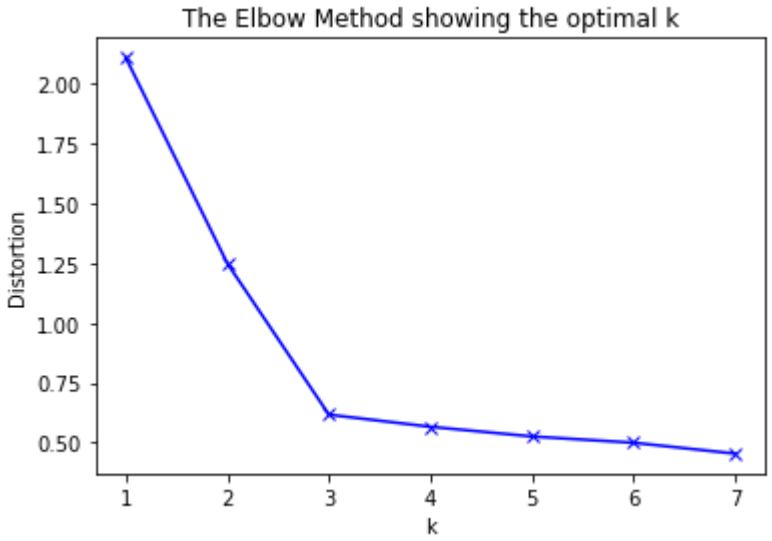
Out[5]:
```
<matplotlib.collections.PathCollection at 0x26e97d2d4e0>
```



In [6]:
```
from sklearn.cluster import KMeans
import numpy as np
```

In [7]:
```
# k means determine k
distortions = []
K = range(1,8)
for k in K:
    kmeanModel = KMeans(n_clusters=k).fit(df)
    kmeanModel.fit(df)
    distortions.append(sum(np.min(cdist(df, kmeanModel.cluster_centers_,
                                        'euclidean'), axis=1)) / df.shape[0])

# 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]:
```
# => Select k = 3
kmeans = KMeans(n_clusters=3)
kmeans.fit(df)

centroids = kmeans.cluster_centers_
labels = kmeans.labels_

print(centroids)
print(labels)
```

```
[[-1.5947298   2.92236966]
 [ 2.06521743  0.96137409]
 [ 0.9329651   4.35420712]]
[1 2 2 2 1 2 2 1 0 2 1 0 0 2 2 0 0 1 0 1 2 1 2 2 0 1 1 2 0 1 0 0 0 0 2 1 1
 1 2 2 0 0 2 1 1 1 0 2 0 2 1 2 2 1 1 0 2 1 0 2 0 0 0 0 2 0 2 1 2 2 2 1 1 2
 1 2 2 0 0 2 1 1 2 2 1 1 1 0 0 1 1 2 1 2 1 2 0 0 1 1 1 1 0 1 1 2 0 2 2 2 0
 2 1 0 2 0 2 2 0 0 2 1 2 2 1 1 0 1 0 0 0 0 1 0 0 0 2 0 1 0 2 2 1 1 0 0 0 0
 1 1]
```

In [9]:
```
df['Group'] = pd.Series(labels)
df.head()
```

Out[9]:

	f1	f2	Group
0	2.605097	1.225296	1
1	0.532377	3.313389	2
2	0.802314	4.381962	2
3	0.528537	4.497239	2
4	2.618585	0.357698	1

In [10]:
```
plt.scatter(centroids[:, 0],centroids[:, 1],
            marker = "s", s=15, color='red')
plt.scatter(df.f1, df.f2, c=df.Group)
plt.show()
```