K Means Clustering

In [1]:

```
#importing libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [29]:

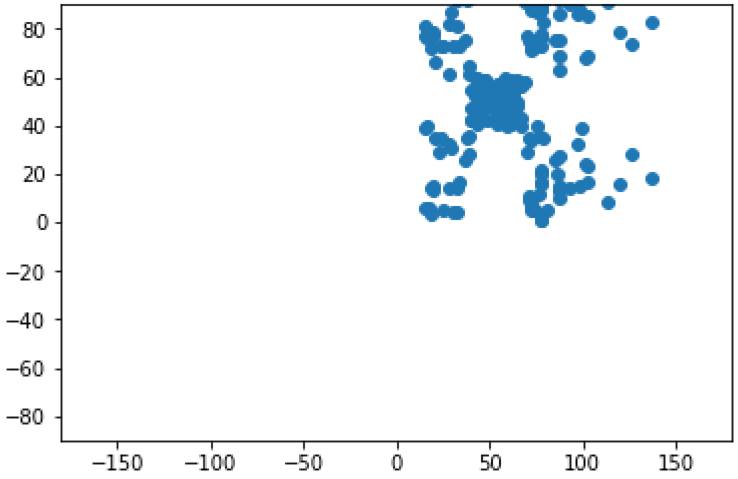
```
#Import mall dataset

dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:,[3,4]].values
dataset.head()
```

Out[29]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

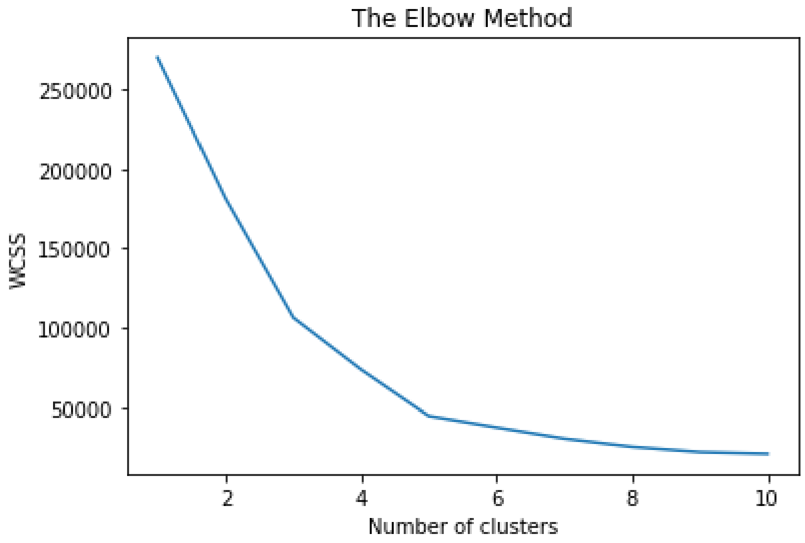
In [28]:

```
plt.scatter(dataset['Annual Income (k$)'],dataset['Spending Score (1-100)'])
plt.xlim(-180,180)
plt.ylim(-90,90)
plt.show()
```



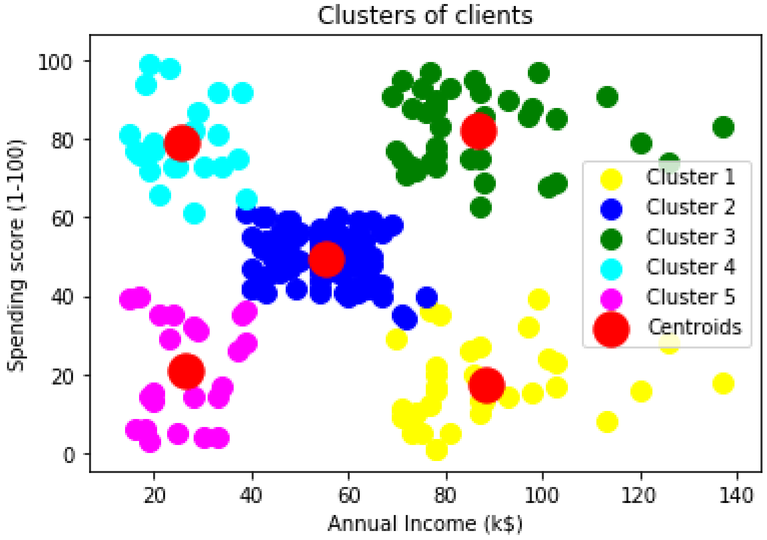
In [24]:

```
#Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters = i, init='k-means++',max_iter = 300, n_init=10, random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

#Plot the graph to visualize the Elbow Metrhod to find the optimal number of cluster
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



In [27]:

```
#applying KMeans to the dataset with the optimal number of clusters
kmeans = KMeans(n_clusters = 5, init='k-means++',max_iter = 300, n_init=10, random_state = 0)
Y_Kmeans = kmeans.fit_predict(X)
#Visualizing the clusters
plt.scatter(X[Y_Kmeans ==0,0],X[Y_Kmeans ==0,1],s=100,c='yellow',label = 'Cluster 1')
plt.scatter(X[Y_Kmeans ==1,0],X[Y_Kmeans ==1,1],s=100,c='blue',label = 'Cluster 2')
plt.scatter(X[Y_Kmeans ==2,0],X[Y_Kmeans ==2,1],s=100,c='green',label = 'Cluster 3')
plt.scatter(X[Y_Kmeans ==3,0],X[Y_Kmeans ==3,1],s=100,c='cyan',label = 'Cluster 4')
plt.scatter(X[Y_Kmeans ==4,0],X[Y_Kmeans ==4,1],s=100,c='magenta',label = 'Cluster 5')
#plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=300,c='yellow',label = 'Centroids')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s = 300, c = 'red', label = 'Centroids')

plt.title('Clusters of clients')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending score (1-100)')
plt.legend()
plt.show()
```



In [ ]: