K-MEANS Clustering

K-means Clustering is an Unsupervised learning algorithm.It performs division of objects into clusters which are similar between them and are dissimilar to the objects belonging to another cluster.

Name: Arathi Unnikrishnan

Task 2:Predicting the optimum numbers of clusters from the Unsupervised dataset and representing it visually

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data,columns=iris.feature_names)
iris_df.head()
```

Out[3]:

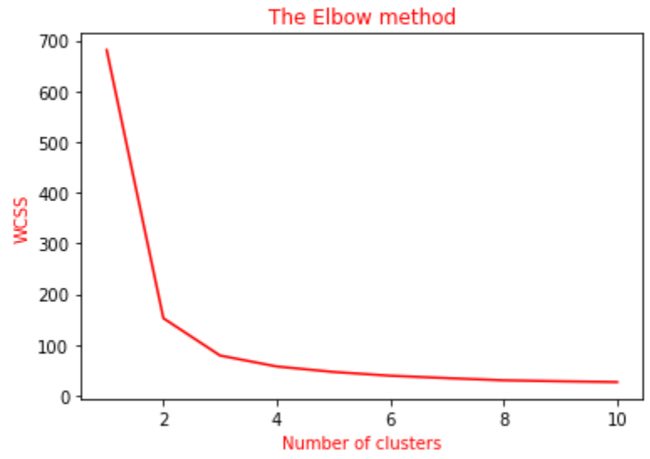
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

In [11]:
```
x = iris_df.iloc[:,[0,1,2,3]].values

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_)

# Plotting the results onto a line graph,
# allowing us to observe 'The elbow'
plt.plot(range(1,11),wcss,color='red')
plt.title('The Elbow method', color='red')
plt.xlabel('Number of clusters',color='red')
plt.ylabel('WCSS',color='red') # Within cluster sum of squares
plt.show()
```



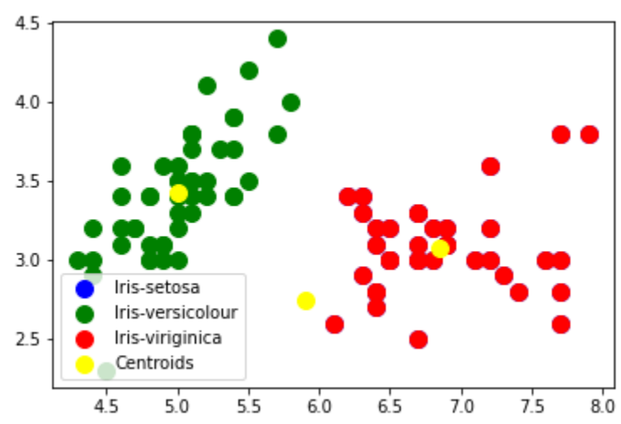
Creating the k-means classifier

In [13]:
```
kmeans = KMeans(n_clusters = 3, init = 'k-means++',
                max_iter = 300, n_init = 10, random_state = 0)
y_kmeans = kmeans.fit_predict(x)
```

Visualising the clusters

In [15]:
```
plt.scatter(x[y_kmeans == 0, 0],x[y_kmeans == 0,1],
            s = 100,c = 'blue', label = 'Iris-setosa')
plt.scatter(x[y_kmeans == 1, 0],x[y_kmeans == 1,1],
            s = 100,c = 'green',label = 'Iris-versicolour')
plt.scatter(x[y_kmeans == 0, 0],x[y_kmeans == 0,1],
            s=100,c = 'red',label = 'Iris-viriginica')
# Plotting the centroids of the clusters
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
            s = 100, c = 'yellow', label = 'Centroids')
plt.legend()
```

Out[15]: <matplotlib.legend.Legend at 0x23493267220>



In [ ]: