In [12]:
```
import pandas as pd
import glob
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

In [13]:
```
#data = pd.read_csv("/home/manu/Downloads/europe11.csv")
import openpyxl # for reading excel file

xls = pd.ExcelFile(r"C:\Users\reshma_koduri\OneDrive\Documents\kmeans.xlsx", engine=
data = pd.read_excel(xls)

#data = pd.read_csv("/home//Desktop/online/fiat500.csv")
```

In [14]:
```
data.describe()
```

Out[14]:

	srno	height	weight
count	10.00000	10.000000	10.000000
mean	5.50000	179.500000	71.400000
std	3.02765	6.186904	9.008638
min	1.00000	168.000000	56.000000
25%	3.25000	179.250000	68.500000
50%	5.50000	180.000000	71.500000
75%	7.75000	182.750000	75.750000
max	10.00000	188.000000	84.000000

In [15]:
```
data
```

Out[15]:

	srno	height	weight
0	1	185	72
1	2	170	56
2	3	168	60
3	4	179	68
4	5	182	72
5	6	188	77
6	7	180	71
7	8	180	70
8	9	183	84
9	10	180	84

In [16]:
```
X=data.drop(['srno'],axis=1)
```

In [17]:
```
#data1.head(10)
X
```

Out[17]:

	height	weight
0	185	72
1	170	56
2	168	60
3	179	68
4	182	72
5	188	77
6	180	71
7	180	70
8	183	84
9	180	84

In [18]:
```
cor_mat= data.corr()
cor_mat
```

Out[18]:

	srno	height	weight
srno	1.000000	0.341072	0.749567
height	0.341072	1.000000	0.759539
weight	0.749567	0.759539	1.000000

In [19]:
```
X
```

Out[19]:

	height	weight
0	185	72
1	170	56
2	168	60
3	179	68
4	182	72
5	188	77
6	180	71
7	180	70
8	183	84
9	180	84

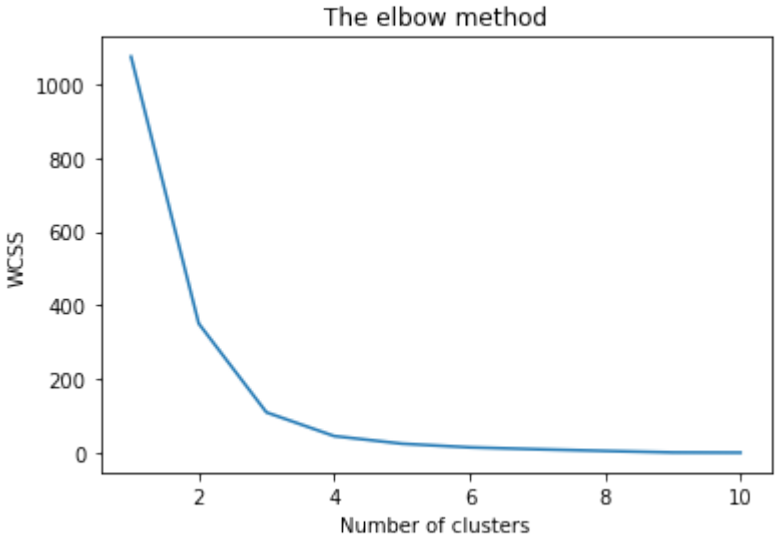
In [ ]:
```

```

In [20]:
```
#To find optimum 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
    kmeans = KMeans(n_clusters = i, max_iter = 300)
    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)
plt.title('The elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS') #within cluster sum of squares
plt.show()
```



In [21]:
```
#sample KMEANS
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters =3)
ypred=kmeans.fit_predict(X)
ypred=kmeans.predict(X)
#ypred
```

In [22]:
```
ypred
```

Out[22]:
```
array([2, 1, 1, 2, 2, 0, 2, 2, 0, 0])
```

In [23]:
```
data['catagory']=ypred
```

In [24]:
```
data.head(30)
```

Out[24]:

	srno	height	weight	catagory
0	1	185	72	2
1	2	170	56	1
2	3	168	60	1

	srno	height	weight	catagory
3	4	179	68	2
4	5	182	72	2
5	6	188	77	0
6	7	180	71	2
7	8	180	70	2
8	9	183	84	0
9	10	180	84	0

In [ ]:

In [ ]: