Vashu Agarwal

E21CSEU0054 EB06 Lab7 Q2

In [8]:
```
import numpy as np
import matplotlib.pyplot as mtp
import pandas as pd
```

In [9]:
```
dataset = pd.read_csv("/Users/vashuagarwal/Downloads/BENNETT  thing
```

In [10]:
```
print(dataset.head(5))
```

```
  private  apps  accept  enroll  top10perc  top25perc  f_undergrad
\
0     Yes  1660    1232     721         23         52         2885
1     Yes  2186    1924     512         16         29         2683
2     Yes  1428    1097     336         22         50         1036
3     Yes   417     349     137         60         89          510
4     Yes   193     146      55         16         44          249

   p_undergrad  outstate  room_board  books  personal  phd  termin
al  \
0          537      7440        3300    450      2200   70
78
1         1227     12280        6450    750      1500   29
30
2           99     11250        3750    400      1165   53
66
3           63     12960        5450    450       875   92
97
4          869      7560        4120    800      1500   76
72

   s_f_ratio  perc_alumni  expend  grad_rate
0       18.1           12    7041         60
1       12.2           16   10527         56
2       12.9           30    8735         54
3        7.7           37   19016         59
4       11.9            2   10922         15
```

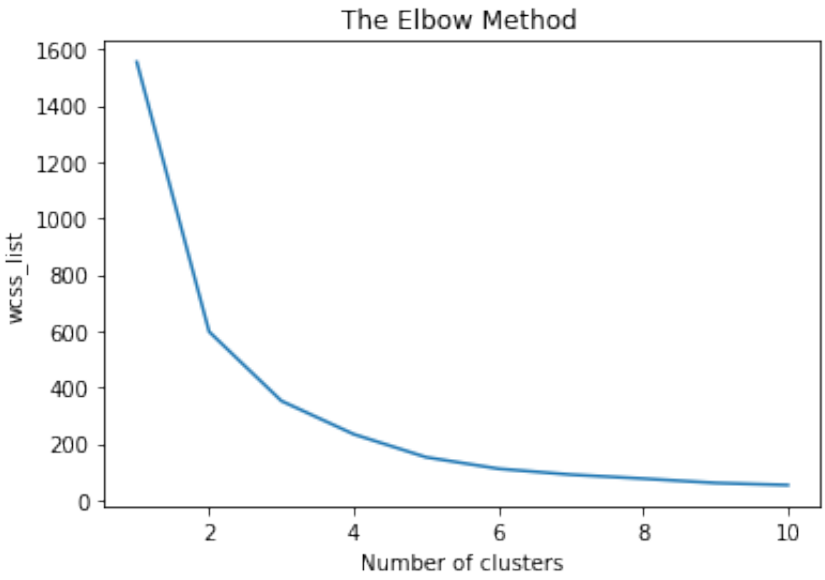
In [57]:
```
x = dataset.iloc[:,[1,2]].values
print(x)
```

```
[[ 1660  1232]
 [ 2186  1924]
 [ 1428  1097]
 ...
 [ 2097  1915]
 [10705  2453]
 [ 2989  1855]]
```

In [58]:
```
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x = st_x.fit_transform(x)
```

In [59]:
```
from sklearn.cluster import KMeans
wcss_list = []
```

In [60]:
```
for i in range(1,11):
    kmeans = KMeans(n_clusters = i,init= 'k-means++',random_state =
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1,11),wcss_list)
mtp.title("The Elbow Method ")
mtp.xlabel("Number of clusters")
mtp.ylabel("wcss_list")
mtp.show()
```

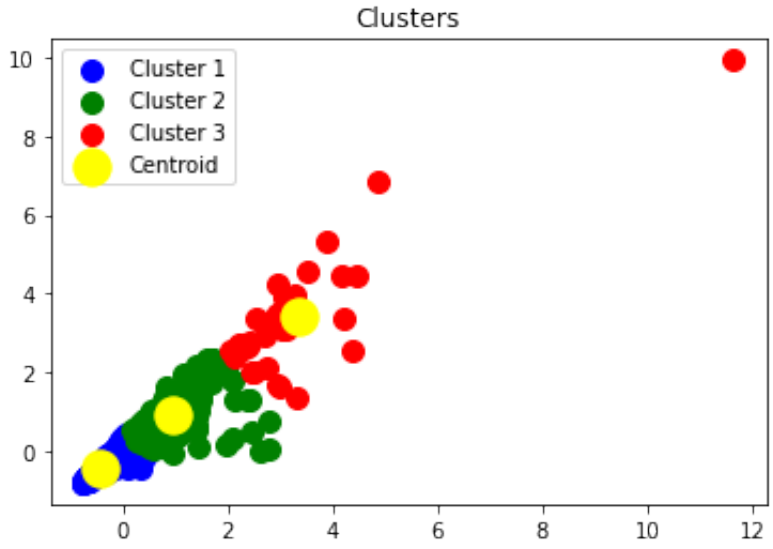


In [ ]:
```
```

In [62]:
```
kmeans = KMeans(n_clusters = 3,init = 'k-means++',random_state = 42
y_predict = kmeans.fit_predict(x)
```

In [ ]:
```
```

In [63]:
```
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c
mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c =
# mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100,
# mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100,
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[
mtp.title("Clusters")
mtp.legend()
mtp.show()
```



In [ ]:
```
```

In [ ]:
```
```

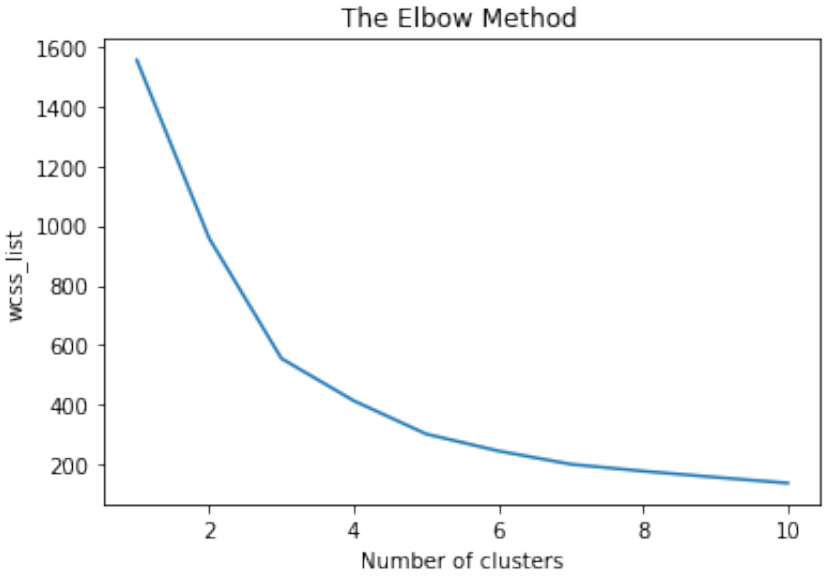
In [74]:
```
x = dataset.iloc[:,[3,4]].values
print(x)
```

```
[[ 721   23]
 [ 512   16]
 [ 336   22]
 ...
 [ 695   34]
 [1317   95]
 [ 691   28]]
```

In [75]:
```
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x = st_x.fit_transform(x)
```

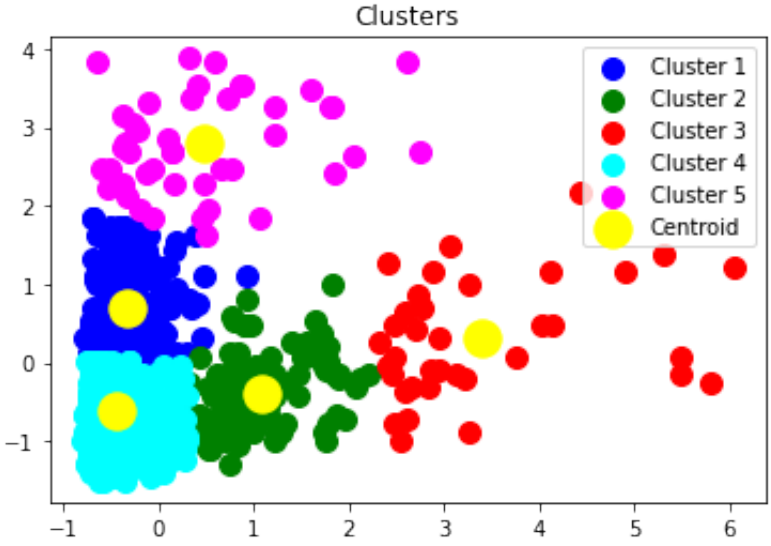
In [76]:
```
from sklearn.cluster import KMeans
wcss_list = []
```

In [77]:
```
for i in range(1,11):
    kmeans = KMeans(n_clusters = i,init= 'k-means++',random_state =
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1,11),wcss_list)
mtp.title("The Elbow Method ")
mtp.xlabel("Number of clusters")
mtp.ylabel("wcss_list")
mtp.show()
```



In [79]:
```
kmeans = KMeans(n_clusters = 5,init = 'k-means++',random_state = 42
y_predict = kmeans.fit_predict(x)
```

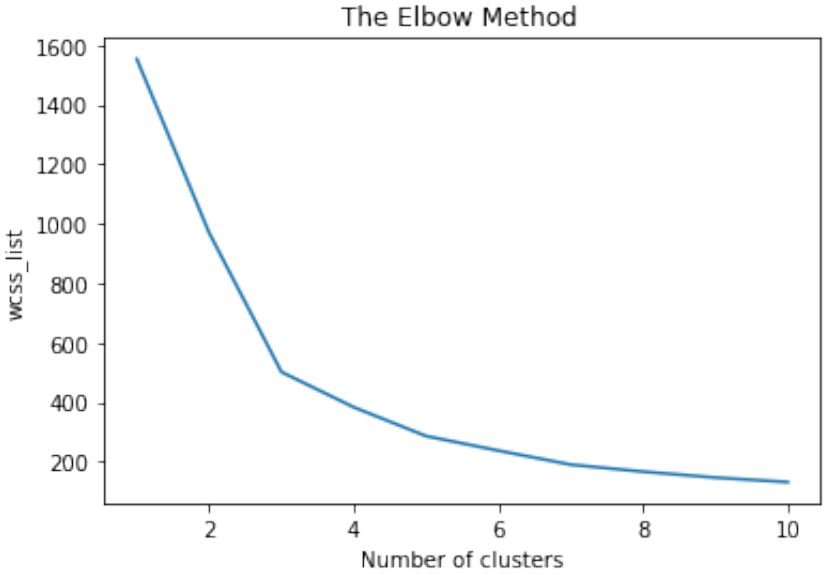
In [80]:
```
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c
mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c =
mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c
mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[
mtp.title("Clusters")
mtp.legend()
mtp.show()
```



In [ ]:

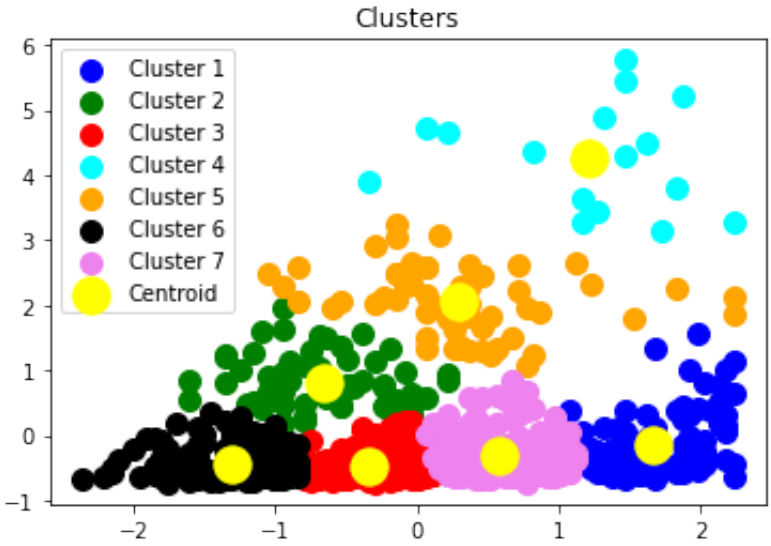
In [81]:
```
x = dataset.iloc[:,[5,6]].values

from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x = st_x.fit_transform(x)
from sklearn.cluster import KMeans
wcss_list = []
for i in range(1,11):
    kmeans = KMeans(n_clusters = i,init= 'k-means++',random_state =
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1,11),wcss_list)
mtp.title("The Elbow Method ")
mtp.xlabel("Number of clusters")
mtp.ylabel("wcss_list")
mtp.show()
```



In [ ]:
```
```

In [82]:
```
kmeans = KMeans(n_clusters = 7,init = 'k-means++',random_state = 42
y_predict = kmeans.fit_predict(x)
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c
mtp.scatter(x[y_predict == 2, 0], x[y_predict == 2, 1], s = 100, c
mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c
mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c
mtp.scatter(x[y_predict == 5, 0], x[y_predict == 5, 1], s = 100, c
mtp.scatter(x[y_predict == 6, 0], x[y_predict == 6, 1], s = 100, c
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[
mtp.title("Clusters")
mtp.legend()
mtp.show()
```



In [ ]:

In [ ]:

In [ ]: