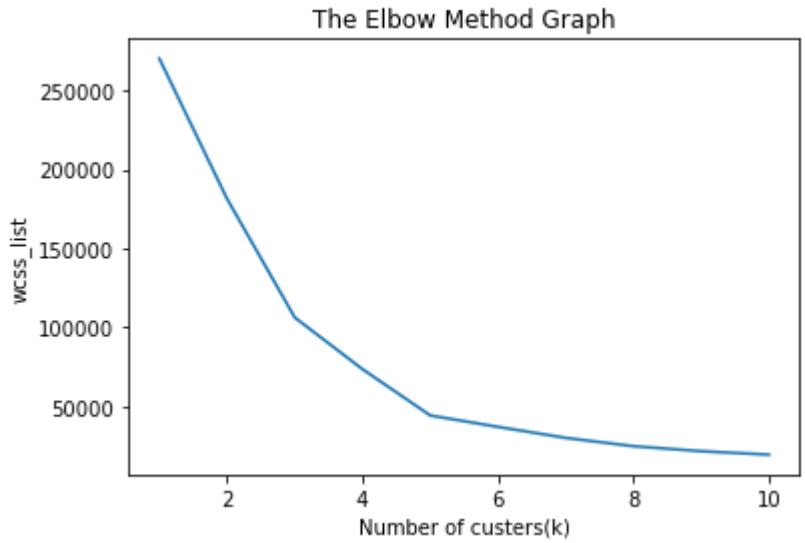
In [1]:
```
#importing Libraries
import numpy as np
import matplotlib.pyplot as mtp
import pandas as pd
```

In [2]:
```
dataset = pd.read_csv('Mall_Customers.csv')
```

In [3]:
```
x= dataset.iloc[:,[3,4]].values
```

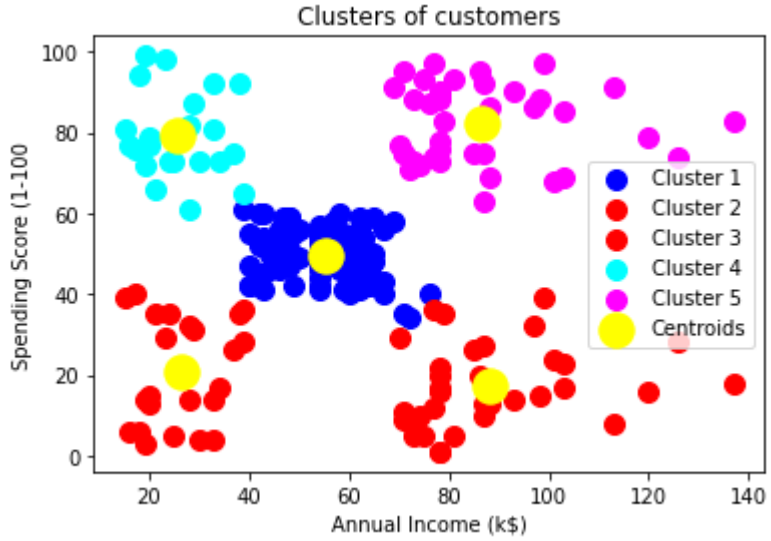
In [4]:
```
#finding optimal number of clusters using the elbow method
from sklearn.cluster import KMeans
wcss_list = [] #initializing the list for the values of WCSS
#using for loop for iterations from 1 to 10
for i in range(1,11):
    kmeans = KMeans(n_clusters=i, init='k-means++',random_state=42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1,11), wcss_list)
mtp.title('The Elbow Method Graph')
mtp.xlabel('Number of custers(k)')
mtp.ylabel('wcss_list')
mtp.show()
```

```
c:\Users\prati\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: Use
rWarning: KMeans is known to have a memory leak on Windows with MKL, when there
are less chunks than available threads. You can avoid it by setting the environ
ment variable OMP_NUM_THREADS=1.
  warnings.warn(
```



In [5]:
```
#training the K-method model on a dataset
kmeans = KMeans(n_clusters=5, init='k-means++',random_state=42)
y_predict=kmeans.fit_predict(x)
```

In [9]:
```
#Visulaizing the clusters
mtp.scatter(x[y_predict == 0,0], x[y_predict == 0,1], s=100, c ='blue', label='Cl
mtp.scatter(x[y_predict == 1,0], x[y_predict == 1,1], s=100, c ='red', label='Clu
mtp.scatter(x[y_predict == 2,0], x[y_predict == 2,1], s=100, c ='red', label='Clu
mtp.scatter(x[y_predict == 3,0], x[y_predict == 3,1], s=100, c ='cyan', label='Cl
mtp.scatter(x[y_predict == 4,0], x[y_predict == 4,1], s=100, c ='magenta', label=
mtp.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=300, c=
label='Centroids')
mtp.title('Clusters of customers')
mtp.xlabel('Annual Income (k$)')
mtp.ylabel('Spending Score (1-100')
mtp.legend()
mtp.show()
```



In [ ]: