In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

```
<frozen importlib._bootstrap>:219: RuntimeWarning: numpy.ufunc size changed, may indicate binary incomp
atibility. Expected 192 from C header, got 216 from PyObject
```

In [2]:

```
df = pd.read_csv('Mall_Customers.csv')
df.head()
```

Out[2]:

	CustomerID	Gender	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 [3]:

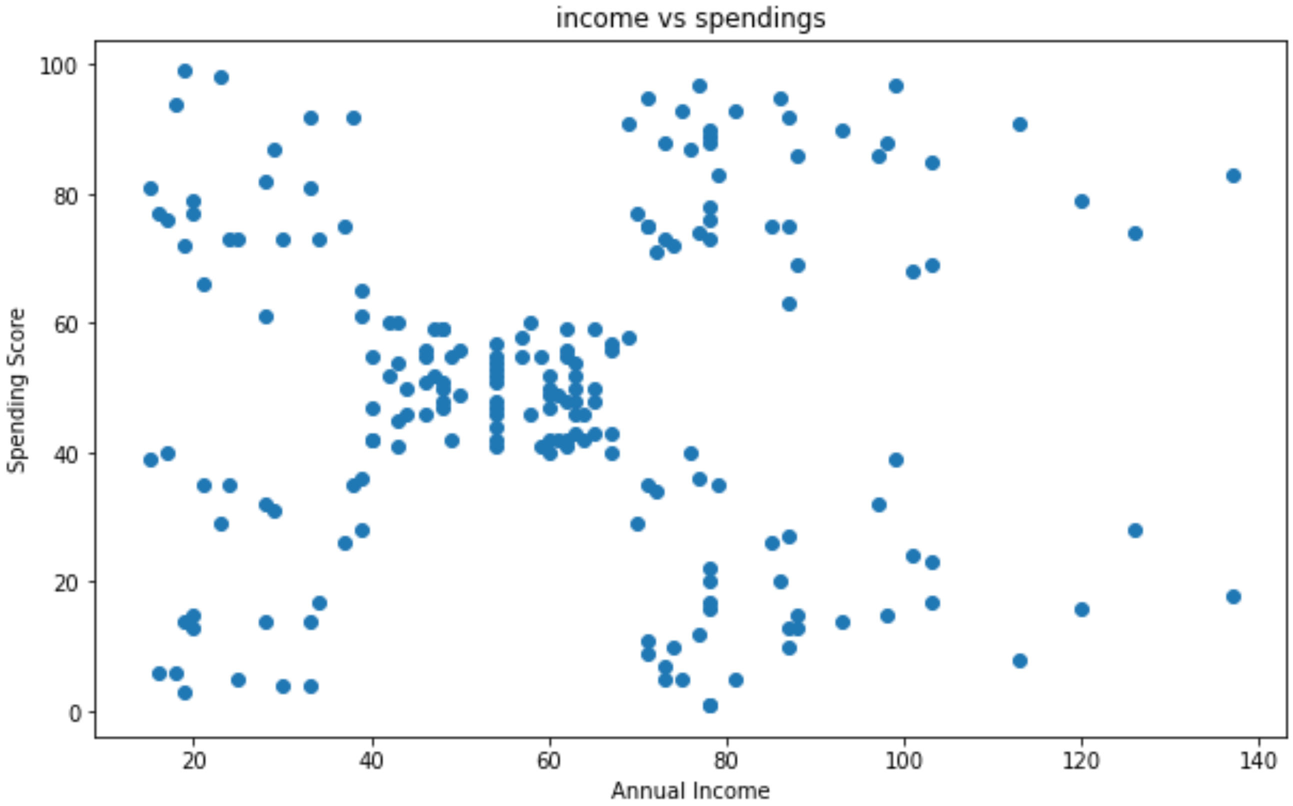
```
df.shape
```

Out[3]:

```
(200, 5)
```

In [4]:

```
plt.figure(figsize = (10,6))
plt.scatter(df['Annual Income (k$)'],df['Spending Score (1-100)'])
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.title("income vs spendings")
plt.show()
```



In [5]:

```
X = df.iloc[:,[3,4]].values
X.shape
```

Out[5]:

```
(200, 2)
```

In [6]:

```
X[:5]
```

Out[6]:

```
array([[15, 39],
       [15, 81],
       [16,  6],
       [16, 77],
       [17, 40]])
```

In [7]:

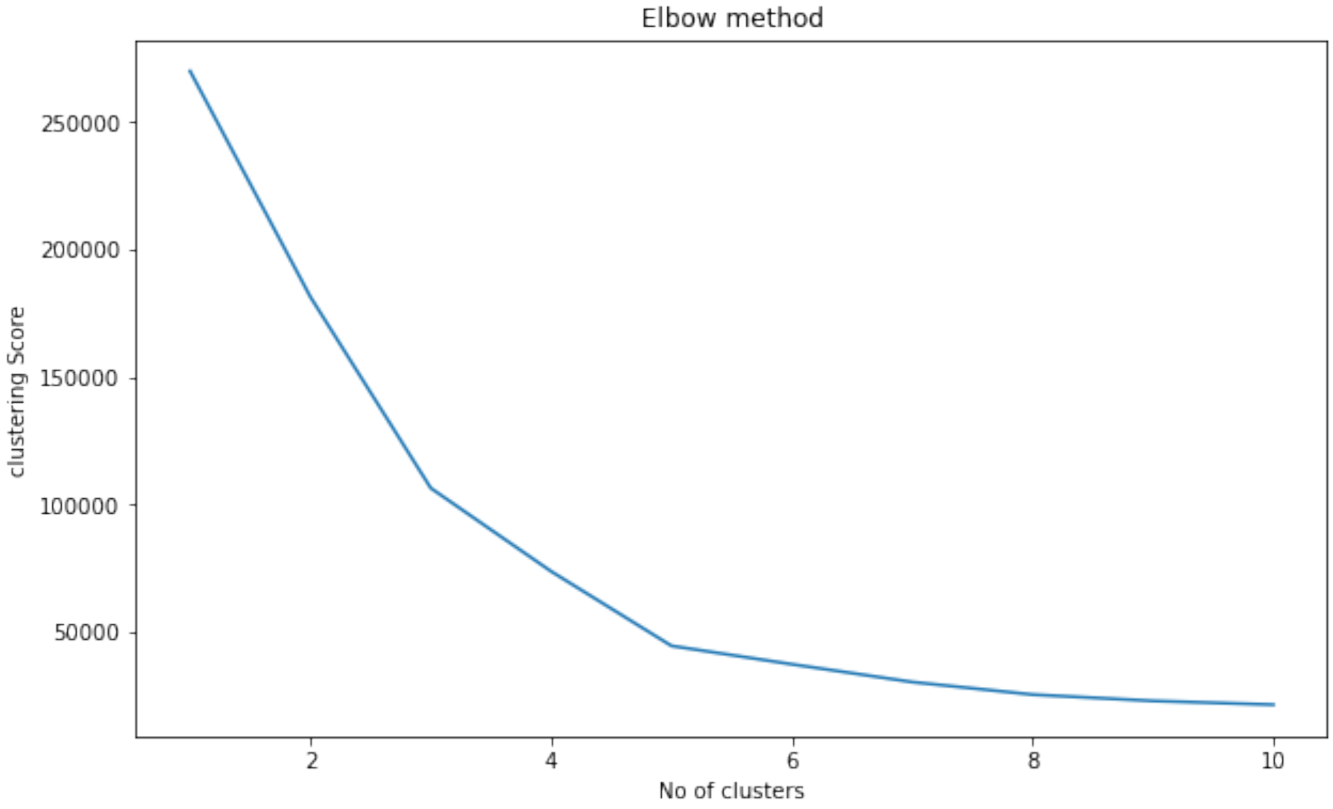
```
# elbow method

clustering_score = []

for i in range(1,11):
    kmeans = KMeans(n_clusters=i,init = 'random',random_state = 42)
    kmeans.fit(X)
    clustering_score.append(kmeans.inertia_)
```

In [10]:

```
plt.figure(figsize = (10,6))
plt.plot(range(1,11),clustering_score)
plt.xlabel('No of clusters')
plt.ylabel('clustering Score')
plt.title("Elbow method")
plt.show()
```



In [13]:

```
clustering_score
```

Out[13]:

```
[269981.28000000014,
 181363.59595959607,
 106348.37306211119,
 73679.78903948837,
 44448.45544793369,
 37233.81451071002,
 30259.657207285458,
 25331.042318281467,
 22855.88239115187,
 21371.67578757027]
```

In [14]:

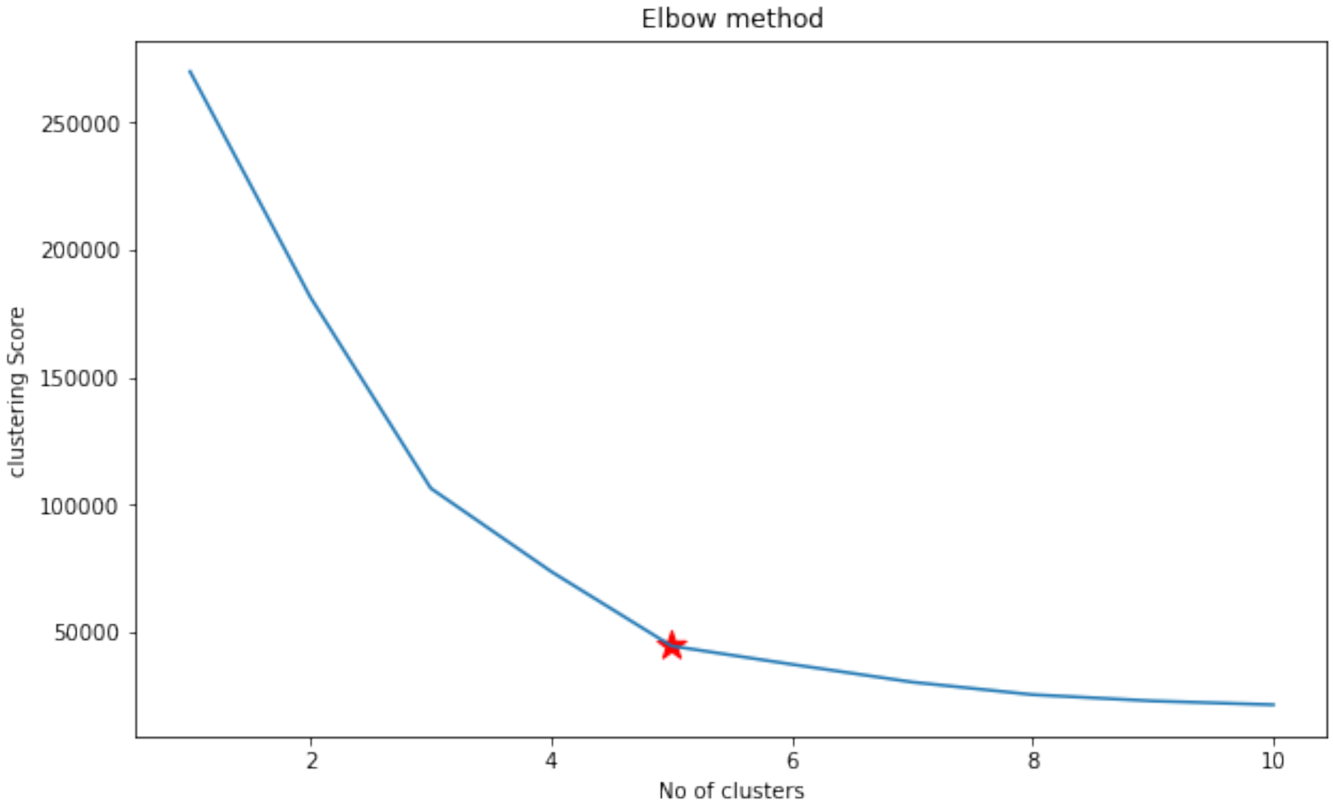
```
clustering_score[4]
```

Out[14]:

```
44448.45544793369
```

In [12]:

```
plt.figure(figsize = (10,6))
plt.plot(range(1,11),clustering_score)
plt.scatter(5,clustering_score[4],s = 200,c = 'red',marker = '*')
plt.xlabel('No of clusters')
plt.ylabel('clustering Score')
plt.title("Elbow method")
plt.show()
```



In [24]:

```python
# silhoutte score : used to determine degree of speration between clusters

# coeff range is in [-1,1]

# if it is 0 : sample is very much closer to neighbouring cluster
# if it is 1 : sample is away from neighbouring cluster
# if it is -1 : sample is assigned to wrong cluster
```

In [21]:

```python
from sklearn.metrics import silhouette_score

silhouette_score_lst = []

for i in range(2,11):
    silhouette_score_lst.append(silhouette_score(X,(KMeans(n_clusters=i).fit_predict(X))))

```

In [22]:

```python
silhouette_score_lst
```

Out[22]:

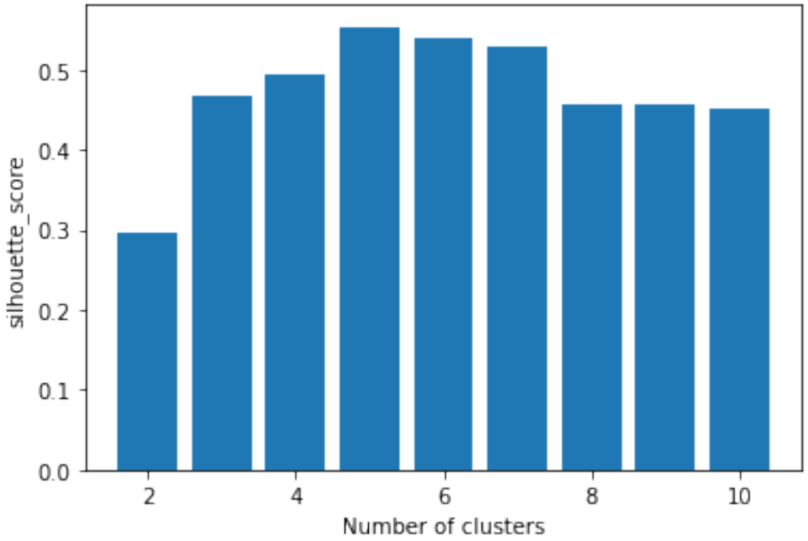
```
[0.2968969162503008,
 0.46761358158775435,
 0.4931963109249047,
 0.553931997444648,
 0.53976103063432,
 0.5288104473798049,
 0.45704384633565154,
 0.457462901394195,
 0.45275118302579015]
```

In [23]:

```
# plotting

k = [2,3,4,5,6,7,8,9,10]

plt.bar(k,silhouette_score_lst)
plt.xlabel("Number of clusters")
plt.ylabel("silhouette_score")
plt.show()
```



In [25]:

```
# highest value of bar from given clusters values will be selected
```

In [26]:

```
# selecting number of clusters = 5
```

In [27]:

```python
# set up a model
kmeans = KMeans(n_clusters=5,random_state = 42)

# fit model
kmeans.fit(X)

# predict
pred = kmeans.predict(X)
print(pred)
```

```
[3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3
 0 3 0 3 0 3 1 3 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 2 4 2 1 2 4 2 4 2 1 2 4 2 4 2 4 2 4 2 1 2 4 2 4 2
 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2 4
 2 4 2 4 2 4 2 4 2 4 2 4 2 4 2]
```

In [30]:

```python
len(pred)
```

Out[30]:

```
200
```

In [28]:

```
df['cluster'] = pd.DataFrame(pred,columns = ['cluster'])
df.head(10)
```

Out[28]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	cluster
0	1	Male	19	15	39	3
1	2	Male	21	15	81	0
2	3	Female	20	16	6	3
3	4	Female	23	16	77	0
4	5	Female	31	17	40	3
5	6	Female	22	17	76	0
6	7	Female	35	18	6	3
7	8	Female	23	18	94	0
8	9	Male	64	19	3	3
9	10	Female	30	19	72	0

In [29]:

```
df['cluster'].value_counts()
```

Out[29]:

```
1    81
2    39
4    35
3    23
0    22
Name: cluster, dtype: int64
```

In [45]:

```
# centroids of each clusters

kmeans.cluster_centers_
```

Out[45]:

```
array([[25.72727273, 79.36363636],
       [55.2962963 , 49.51851852],
       [86.53846154, 82.12820513],
       [26.30434783, 20.91304348],
       [88.2       , 17.11428571]])
```

In [53]:

```
kmeans.cluster_centers_[:,0]
```

Out[53]:

```
array([25.72727273, 55.2962963 , 86.53846154, 26.30434783, 88.2       ])
```

In [38]:

```
X[pred==0,0]
```

Out[38]:

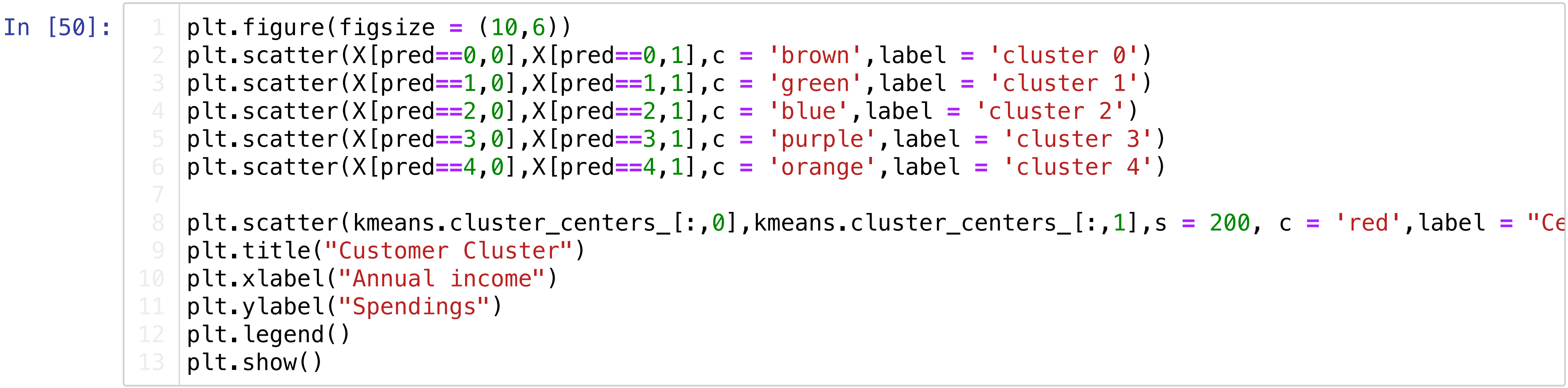
```
array([15, 16, 17, 18, 19, 19, 20, 20, 21, 23, 24, 25, 28, 28, 29, 30, 33,
       33, 34, 37, 38, 39])
```

In [39]:

```
X[pred==0,1]
```

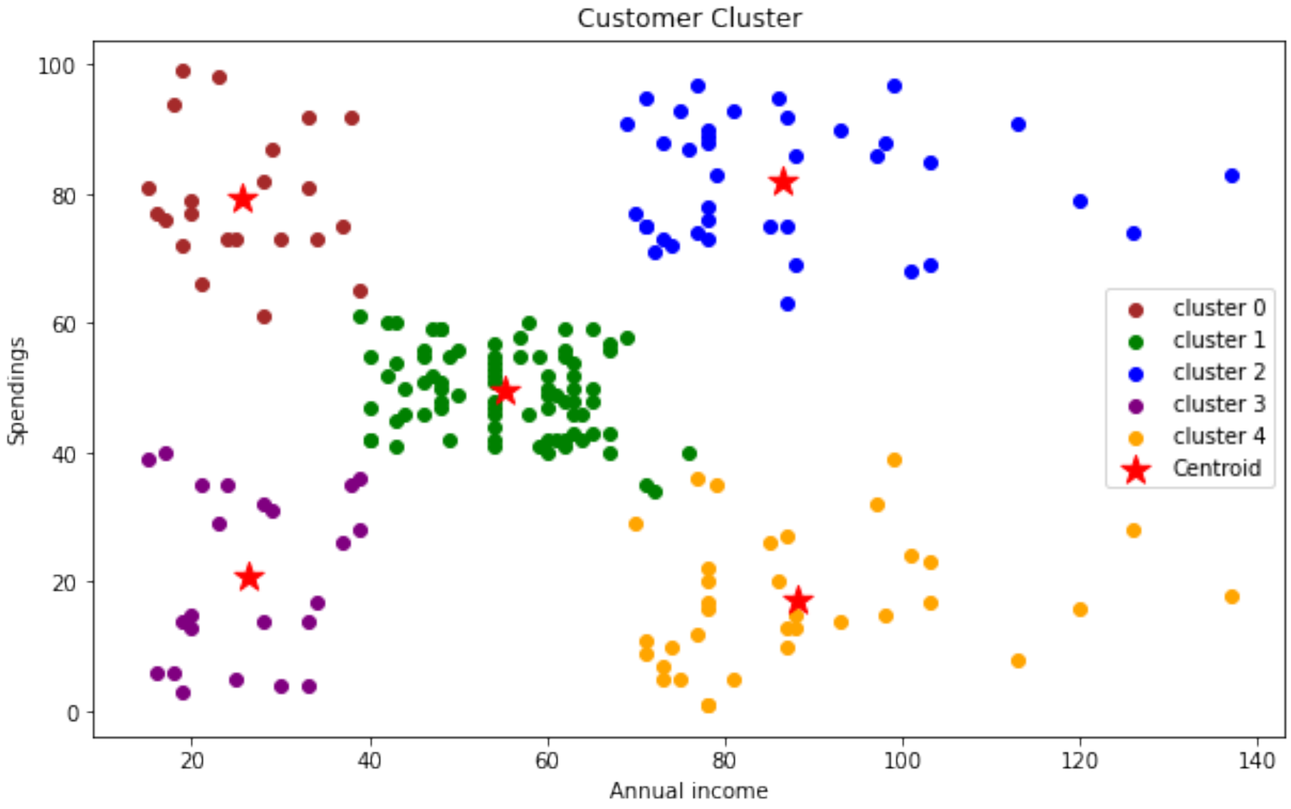
Out[39]:

```
array([81, 77, 76, 94, 72, 99, 77, 79, 66, 98, 73, 73, 82, 61, 87, 73, 92,
       81, 73, 75, 92, 65])
```



```
In [50]:
plt.figure(figsize = (10,6))
plt.scatter(X[pred==0,0],X[pred==0,1],c = 'brown',label = 'cluster 0')
plt.scatter(X[pred==1,0],X[pred==1,1],c = 'green',label = 'cluster 1')
plt.scatter(X[pred==2,0],X[pred==2,1],c = 'blue',label = 'cluster 2')
plt.scatter(X[pred==3,0],X[pred==3,1],c = 'purple',label = 'cluster 3')
plt.scatter(X[pred==4,0],X[pred==4,1],c = 'orange',label = 'cluster 4')

plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s = 200, c = 'red',label = "Ce
plt.title("Customer Cluster")
plt.xlabel("Annual income")
plt.ylabel("Spendings")
plt.legend()
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```