In [54]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [55]:
```
df = pd.read_csv('Mall_Customers.csv')
```

In [56]:
```
df
```

Out[56]:

	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
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

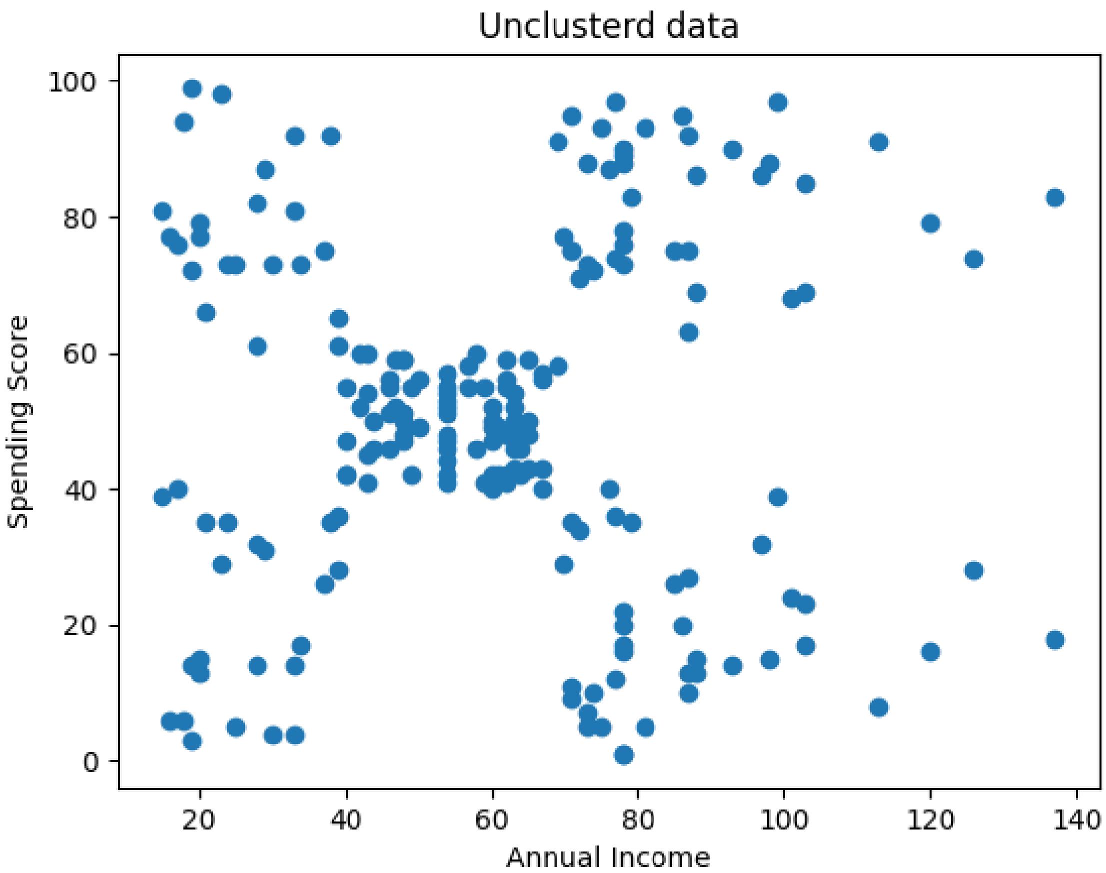
In [57]:
```
x = df.iloc[:,3:]
x
```

Out[57]:

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

In [58]:
```
plt.title("Unclusterd data")
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(df['Annual Income (k$)'], df['Spending Score (1-100)'])
plt.show()
```



In [59]:
```
from sklearn.cluster import KMeans, AgglomerativeClustering
km = KMeans(n_clusters=3)
```

In [60]:
```
x.shape
```

Out[60]:
```
(200, 2)
```

In [61]:
```
km.fit_predict(x)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

Out[61]:
```
array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1], dtype=int32)
```

In [62]:
```
#SSE
km.inertia_
```

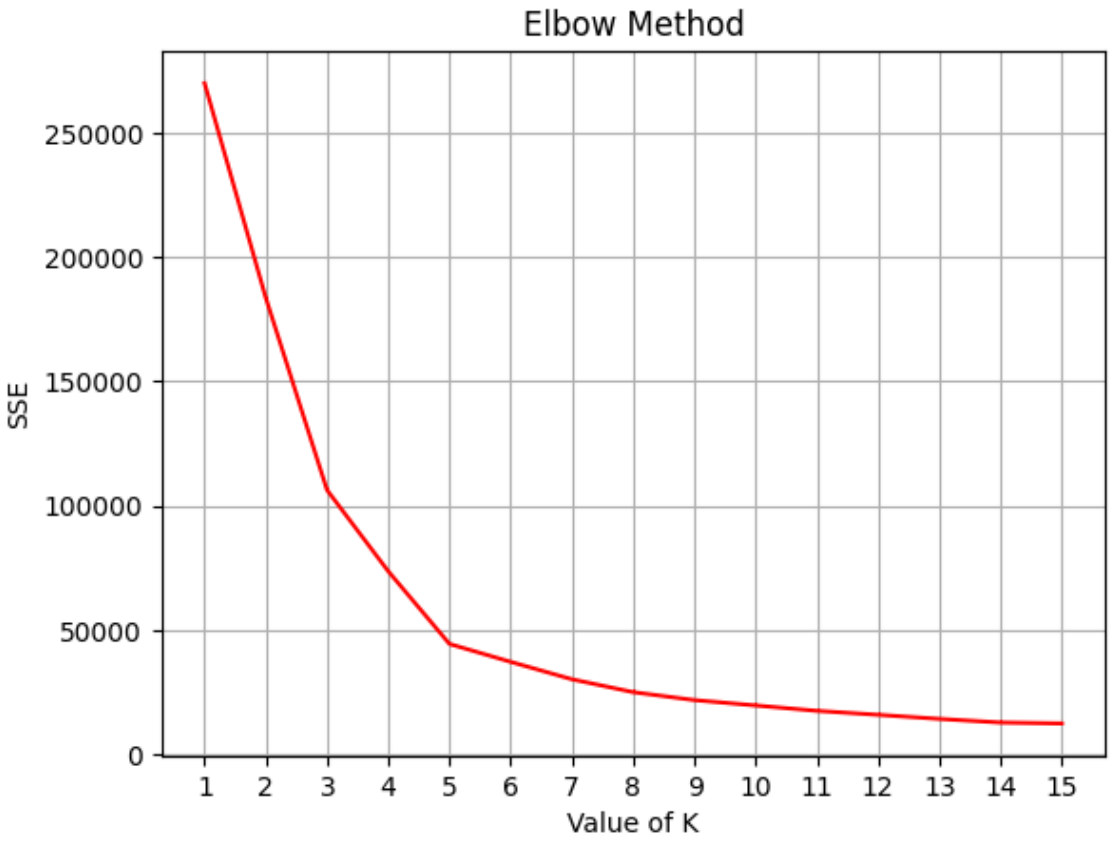
Out[62]:
```
106348.37306211122
```

In [ ]:
```
sse = []
for k in range(1,16):
  km = KMeans(n_clusters=k)
  km.fit_predict(x);
  sse.append(km.inertia_)
```

In [64]:
```
sse
```

Out[64]:
```
[269981.28,
 183653.32894736843,
 106348.37306211122,
 73679.78903948836,
 44448.4554479337,
 37239.835542456036,
 30241.34361793658,
 25062.433792653777,
 21862.092672182895,
 19712.851860217077,
 17546.92800004654,
 15951.036030994095,
 14306.837671530391,
 12844.163231376464,
 12473.186684411685]
```

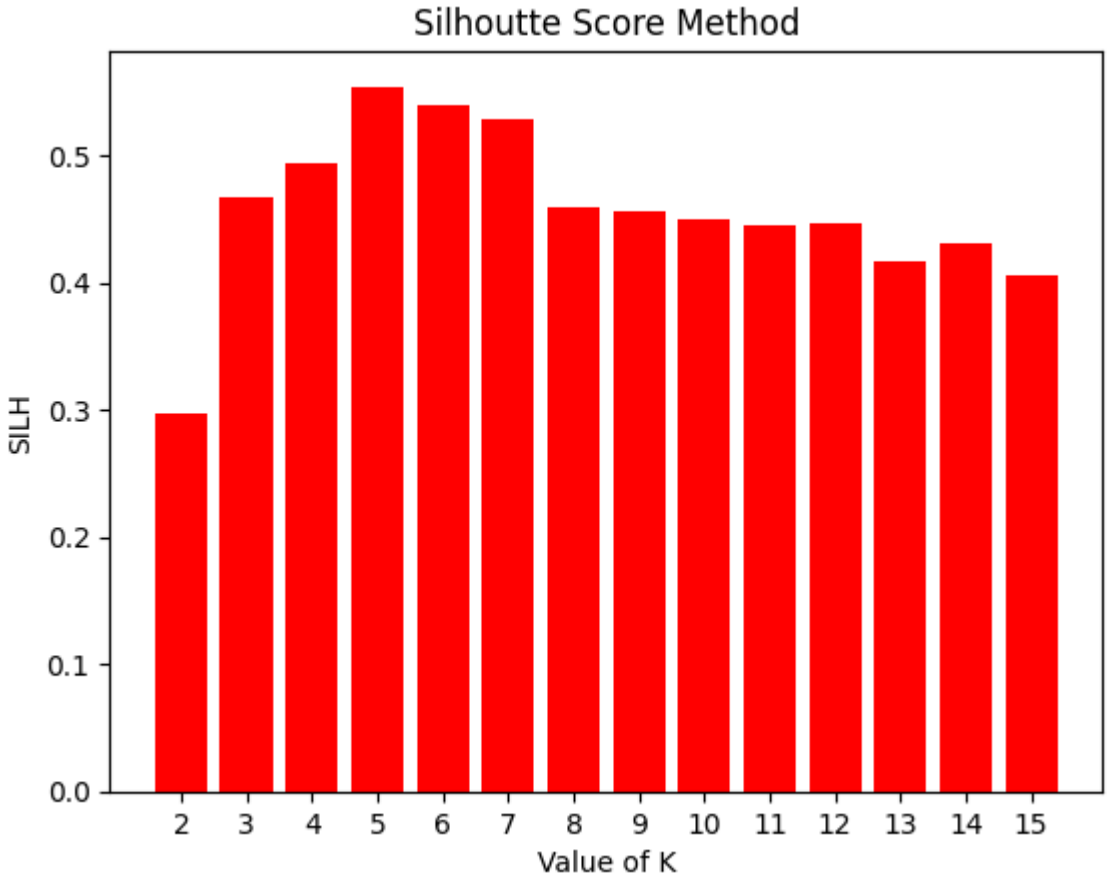
In [65]:
```
plt.title('Elbow Method')
plt.xlabel('Value of K')
plt.ylabel('SSE')
plt.grid()
plt.xticks(range(1,16))
plt.plot(range(1,16), sse, marker=',',color='red')
plt.show()
```



In [66]:
```
from sklearn.metrics import silhouette_score
```

In [ ]:
```
silh = []
for k in range(2,16):
  km = KMeans(n_clusters=k)
  labels = km.fit_predict(x);
  score = silhouette_score(x,labels)
  silh.append(score)
```

In [68]:
```
plt.title('Silhoutte Score Method')
plt.xlabel('Value of K')
plt.ylabel('SILH')
plt.xticks(range(2,16))
plt.bar(range(2,16), silh,color='red')
plt.show()
```



In [69]:
```
km = KMeans(n_clusters=5, random_state=0)
```

In [70]:
```
labels = km.fit_predict(x)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: F
utureWarning: The default value of `n_init` will change from 10 to 'auto'
in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
```

In [71]:
```
labels
```

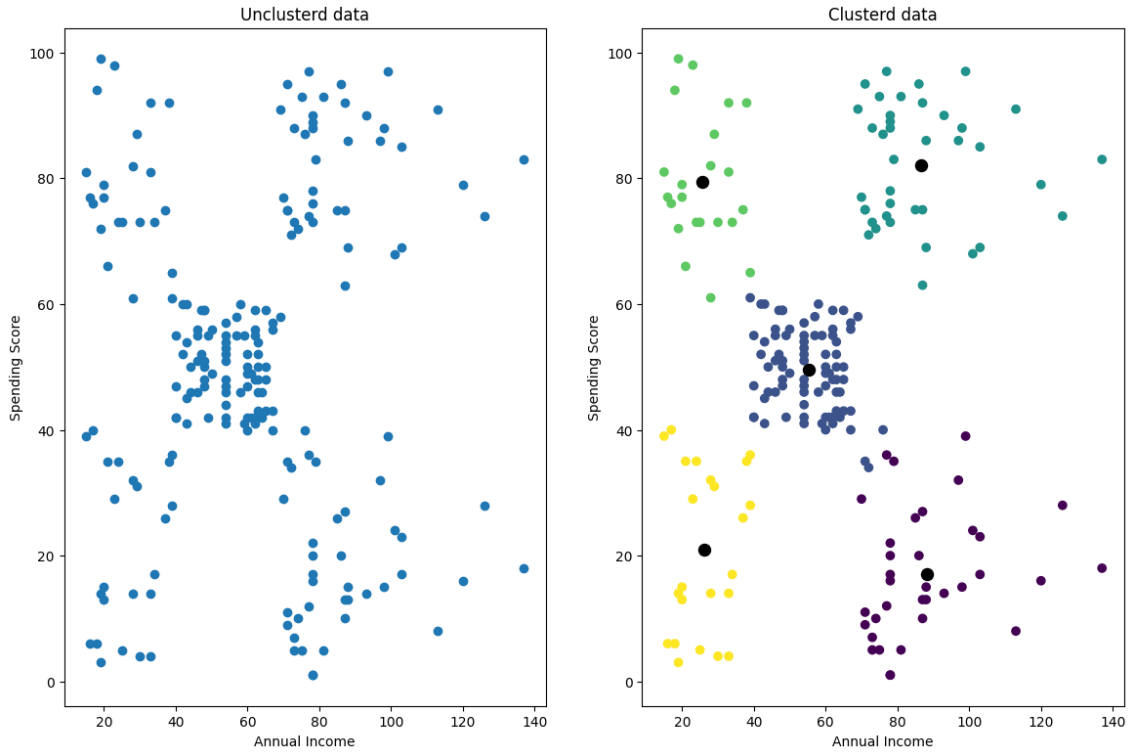
Out[71]:
```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       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, 0, 2, 1, 2, 0, 2, 0, 2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2], dtype=int32)
```

In [72]:
```
cent = km.cluster_centers_
```

In [73]:
```
plt.figure(figsize=(14,9))
plt.subplot(1,2,1)
plt.title("Unclusterd data")
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(df['Annual Income (k$)'], df['Spending Score (1-100)'])

plt.subplot(1,2,2)
plt.title("Clusterd data")
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(df['Annual Income (k$)'], df['Spending Score (1-100)'], c=label
plt.scatter(cent[:,0],cent[:,1], s=75,color='k')
```

Out[73]:
```
<matplotlib.collections.PathCollection at 0x79b92df89a20>
```



In [74]:
```
km.inertia_
```

Out[74]:
```
44448.4554479337
```

In [75]:
```
four = df[labels==4]
```

In [76]:
```
#predict
km.predict([[46,78]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:
X does not have valid feature names, but KMeans was fitted with feature n
ames
  warnings.warn(
```

Out[76]:
```
array([3], dtype=int32)
```

In [77]:
```
agl = AgglomerativeClustering(n_clusters=5)
```

In [78]:
```
alabels = agl.fit_predict(x)
```

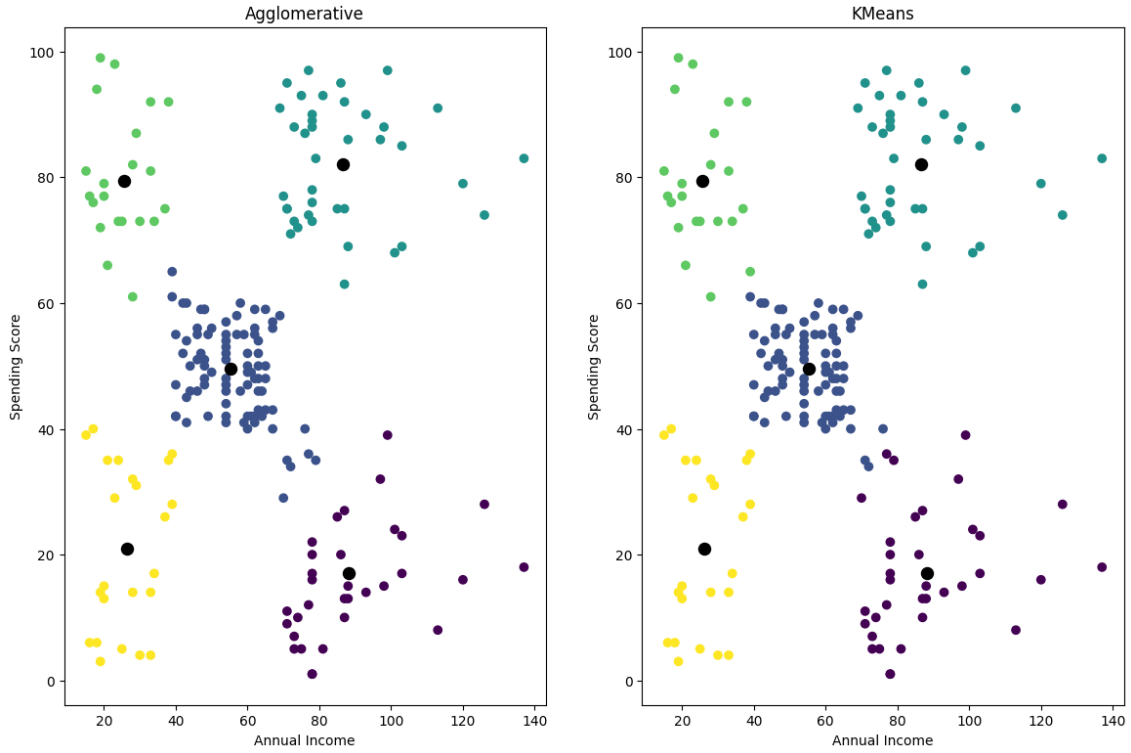
In [79]:
```
alabels
```

Out[79]:
```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 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, 1, 2, 1, 2, 0, 2, 0, 2,
       1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 1, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2])
```

In [80]:
```
plt.figure(figsize=(14,9))
plt.subplot(1,2,1)
plt.title("Agglomerative")
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(df['Annual Income (k$)'], df['Spending Score (1-100)'],
            c=alabels)
plt.scatter(cent[:,0],cent[:,1], s=75,color='k')

plt.subplot(1,2,2)
plt.title("KMeans")
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(df['Annual Income (k$)'], df['Spending Score (1-100)'], c=label
plt.scatter(cent[:,0],cent[:,1], s=75,color='k')
```

Out[80]: <matplotlib.collections.PathCollection at 0x79b92de7fcd0>



In [80]: