In [1]:

```
#clustering Algorithm
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

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

In [3]:

```
df
```

Out[3]:

	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 [4]:

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

In [5]:

```
*x
```

Out[5]:

	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

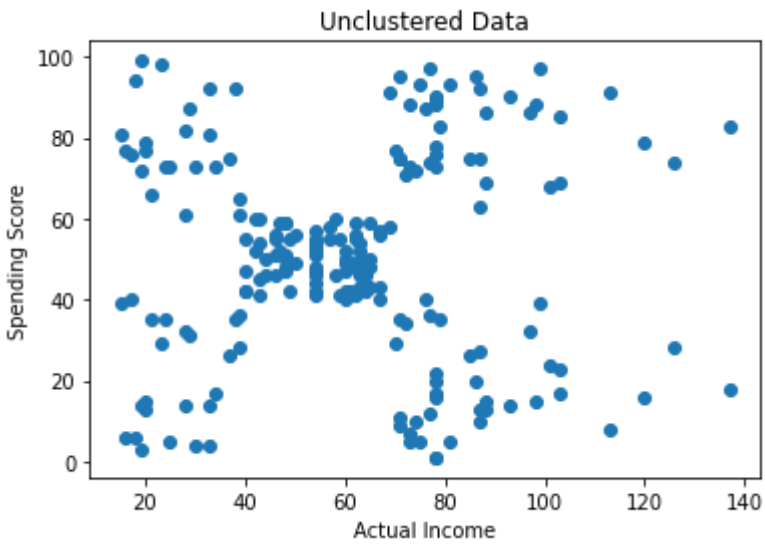
200 rows × 2 columns

In [7]:

```
plt.title('Unclustered Data')
plt.xlabel('Actual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'])
```

Out[7]:

```
<matplotlib.collections.PathCollection at 0x7fd6c51b1790>
```



In [8]:

```
from sklearn.cluster import KMeans, AgglomerativeClustering
```

In [13]:

```
km = KMeans(n_clusters=6)
```

In [14]:

```
km.fit_predict(x)
```

Out[14]:

```
array([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,
0,
       4, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1, 3, 0, 3, 1, 3, 1,
3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 0, 3, 1, 3, 1, 3, 1, 3, 1,
3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1,
3,
       1, 3, 1, 3, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1, 5, 1,
5,
       1, 5], dtype=int32)
```

In [15]:

```
x.shape
```

Out[15]:

```
(200, 2)
```

In [16]:

```
#Sum Squared Error
km.inertia_
```

Out[16]:

```
37239.83554245604
```

In [17]:

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

In [18]:

```
sse
```

Out[18]:

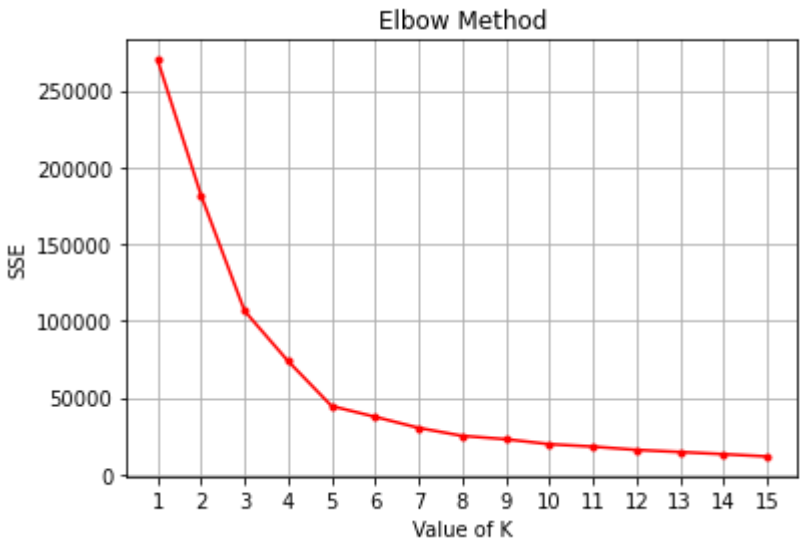
```
[269981.28,
 181665.8231292517,
 106348.37306211118,
 73679.78903948834,
 44448.45544793371,
 37442.24745037571,
 30227.606513152015,
 25061.304119069337,
 22870.41261091155,
 19646.482018947238,
 18057.992060136246,
 15933.528627286692,
 14534.349559295859,
 13144.923153806976,
 11631.774380504554]
```

In [21]:

```
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')
```

Out[21]:

```
[<matplotlib.lines.Line2D at 0x7fd6d02e0340>]
```



In [22]:

```
from sklearn.metrics import silhouette_score
```

In [23]:

```
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 [24]:

```
silh
```

Out[24]:

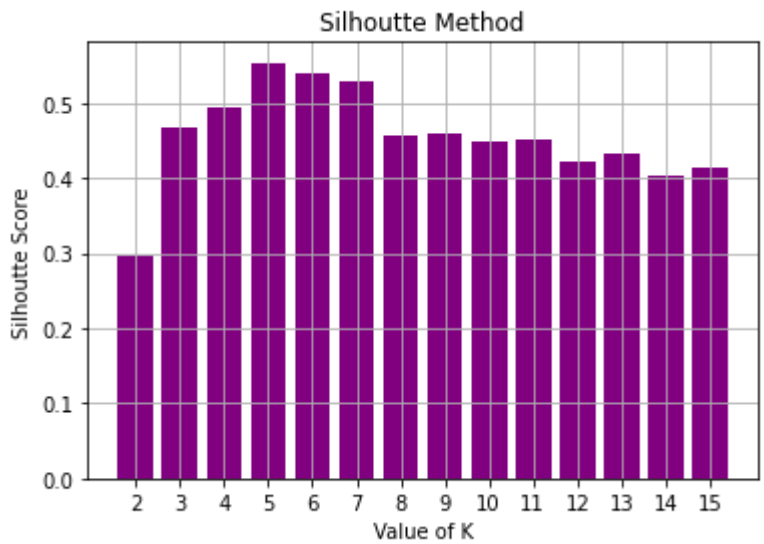
```
[0.2968969162503008,
 0.46761358158775435,
 0.4931963109249047,
 0.553931997444648,
 0.53976103063432,
 0.5288104473798049,
 0.4563394686110682,
 0.45912667975312715,
 0.44760979994374317,
 0.45160313757279347,
 0.42138644395371794,
 0.4319868737519759,
 0.40466460708668867,
 0.4151431541644325]
```

In [32]:

```
plt.title('Silhoutte Method')
plt.xlabel('Value of K')
plt.ylabel('Silhoutte Score')
plt.grid()
plt.xticks(range(2,16))
plt.bar(range(2,16), silh,color='purple')
```

Out[32]:

```
<BarContainer object of 14 artists>
```



In [36]:

```
km = KMeans(n_clusters=5,random_state=0)
```

In [37]:

```
labels = km.fit_predict(x)
```

In [38]:

```
labels
```

Out[38]:

```
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, 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, 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], dtype=int32)
```

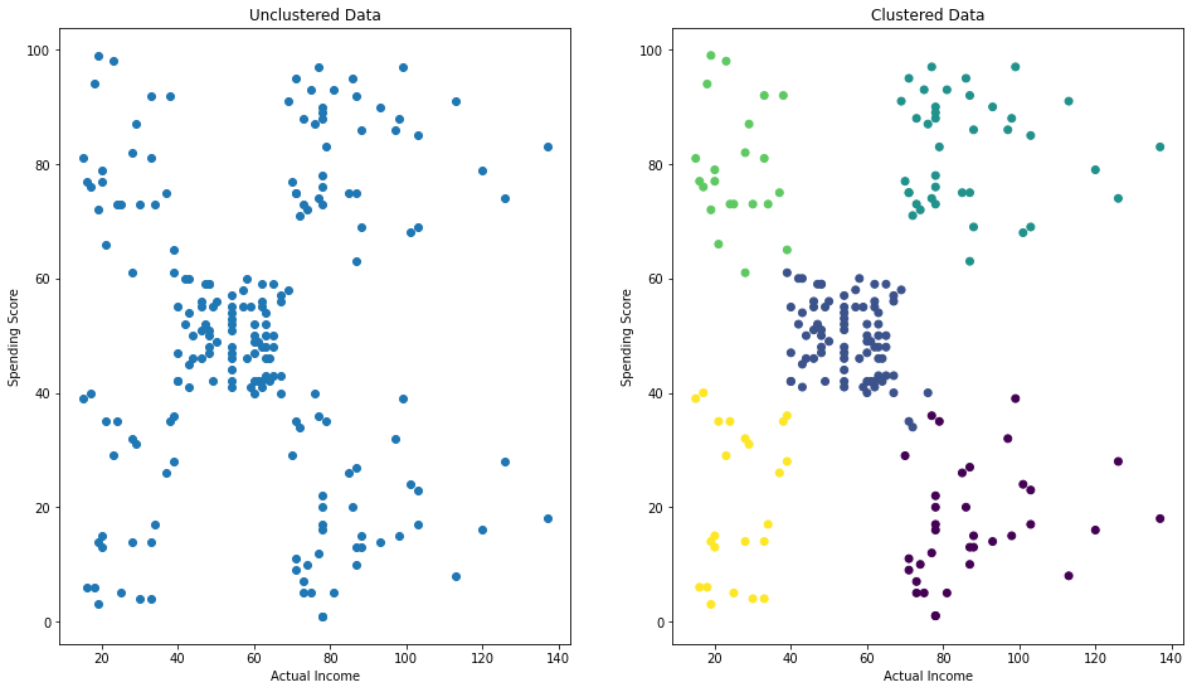
In [41]:

```
plt.figure(figsize=(16,9))
plt.subplot(1,2,1)
plt.title('Unclustered Data')
plt.xlabel('Actual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'])

plt.subplot(1,2,2)
plt.title('Clustered Data')
plt.xlabel('Actual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'],
            c=labels)
```

Out[41]:

```
<matplotlib.collections.PathCollection at 0x7fd6c7652cd0>
```



In [42]:

```
km.cluster_centers_
```

Out[42]:

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

In [43]:

```
cent = km.cluster_centers_
```

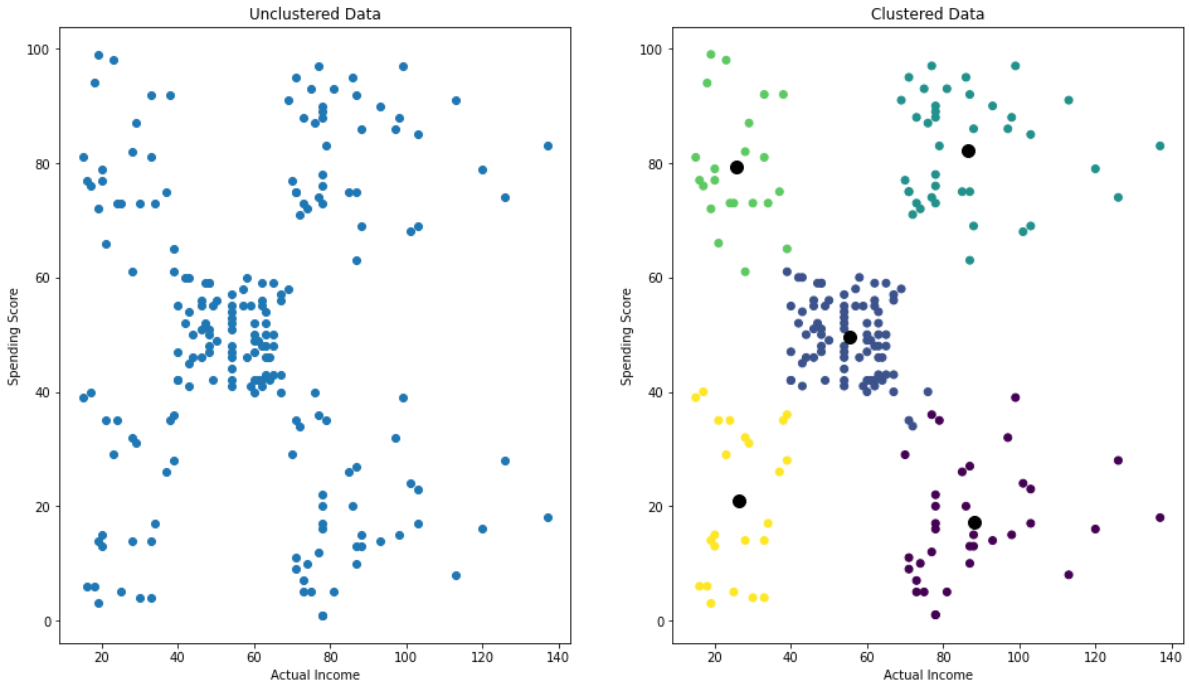
In [44]:

```
plt.figure(figsize=(16,9))
plt.subplot(1,2,1)
plt.title('Unclustered Data')
plt.xlabel('Actual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'])

plt.subplot(1,2,2)
plt.title('Clustered Data')
plt.xlabel('Actual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'],
            c=labels)
plt.scatter(cent[:,0], cent[:,1], s=100,color='k')
```

Out[44]:

```
<matplotlib.collections.PathCollection at 0x7fd6d0642b50>
```



In [45]:

```
km.inertia_
```

Out[45]:

```
44448.45544793371
```

In [46]:

```
km.labels_
```

Out[46]:

```
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, 3,
       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, 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 [51]:

```
df[labels==4]
```

Out[51]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
2	3	Female	20	16	6
4	5	Female	31	17	40
6	7	Female	35	18	6
8	9	Male	64	19	3
10	11	Male	67	19	14
12	13	Female	58	20	15
14	15	Male	37	20	13
16	17	Female	35	21	35
18	19	Male	52	23	29
20	21	Male	35	24	35
22	23	Female	46	25	5
24	25	Female	54	28	14
26	27	Female	45	28	32
28	29	Female	40	29	31
30	31	Male	60	30	4
32	33	Male	53	33	4
34	35	Female	49	33	14
36	37	Female	42	34	17
38	39	Female	36	37	26
40	41	Female	65	38	35
42	43	Male	48	39	36
44	45	Female	49	39	28

In [52]:

```
four = df[labels==4]
```

In [53]:

```
four.to_csv('mydata.csv')
```

In [54]:

```
km.predict([[24,23]])
```

```
/Users/shivraj/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.
py:450: UserWarning: X does not have valid feature names, but KMeans w
as fitted with feature names
  warnings.warn(
```

Out[54]:

```
array([4], dtype=int32)
```

In [55]:

```
km.predict([[34,21]])
```

```
/Users/shivraj/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.
py:450: UserWarning: X does not have valid feature names, but KMeans w
as fitted with feature names
  warnings.warn(
```

Out[55]:

```
array([4], dtype=int32)
```

In [56]:

```
agl = AgglomerativeClustering(n_clusters=5)
```

In [57]:

```
alabels = agl.fit_predict(x)
```

In [58]:

```
alabels
```

Out[58]:

```
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, 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])
```

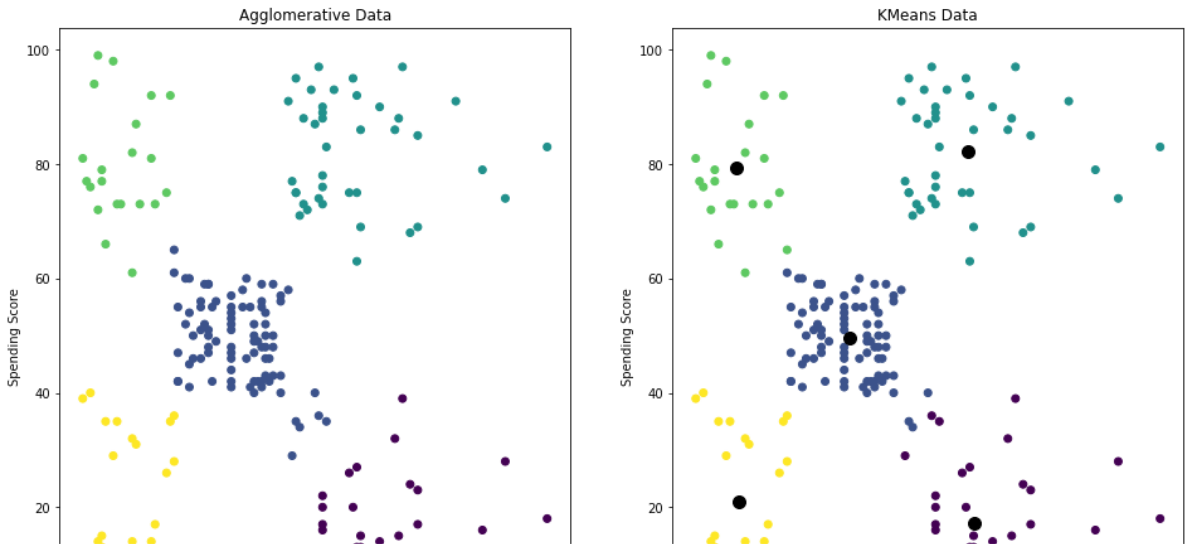
In [60]:

```
plt.figure(figsize=(16,9))
plt.subplot(1,2,1)
plt.title('Agglomerative Data')
plt.xlabel('Actual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'],
            c=alabels)

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

Out[60]:

```
<matplotlib.collections.PathCollection at 0x7fd6b3f30610>
```



In [ ]: