In [1]:
```
# Akanksha Indalkar
#TE IT B 37024
import os
os.getcwd()
```

Out[1]: 'C:\\Users\\AKANKSHA'

In [3]:
```
import os
os.chdir('desktop')
```

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [6]:
```
df = pd.read_csv('Mall_Customers.csv')
```

In [7]:
```
x=df.iloc[ : , 3:]
```

In [10]:
```
#import clustering class of k means
from sklearn.cluster import KMeans, AgglomerativeClustering
km=KMeans(n_clusters=3)
km.fit_predict(x)
```

Out[10]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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, 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, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2,
       0, 2])
```

In [13]:
```
#Find the  best suited value if K ie the number of clusters using elbow method
#sse: sum squared mean We are going to find sse for 1-15 k values and the point at whic

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

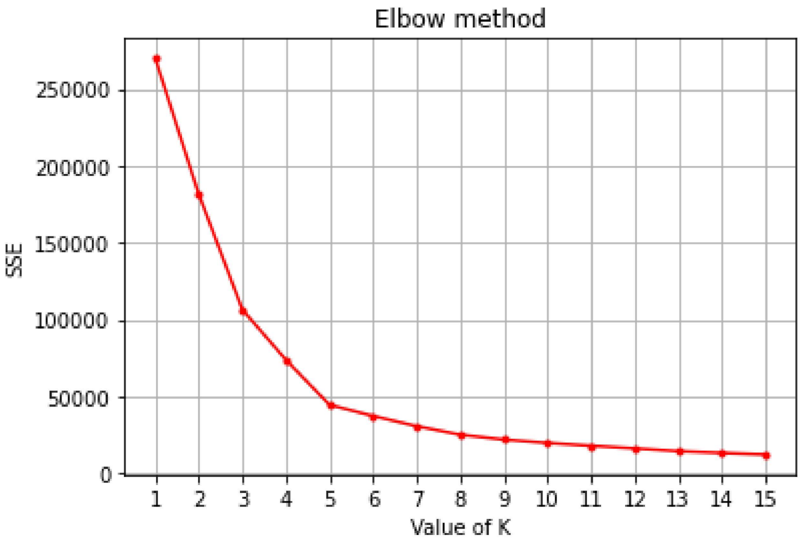
In [14]:
```
sse
```

Out[14]:
```
[269981.28000000014,
 181363.59595959607,
 106348.37306211119,
 73679.78903948837,
 44448.45544793369,
 37239.83554245604,
 30566.45113025185,
 25028.020475269397,
 21850.16528258562,
 19653.383606248837,
 17837.68975468976,
 16132.816658862963,
 14292.543823365135,
```

```
 13164.202123145664,
 12114.285233064027]
```

In [18]:
```
#plot to find the elbow point
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[18]:
```
[<matplotlib.lines.Line2D at 0x2410973e910>]
```



In [21]:
```
#Elbow point is 5. Therefore Value of K=5
km=KMeans(n_clusters=5)
labels=km.fit_predict(x)
```

In [22]:
```
labels
```

Out[22]:
```
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, 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])
```

In [23]:
```
df.columns
```

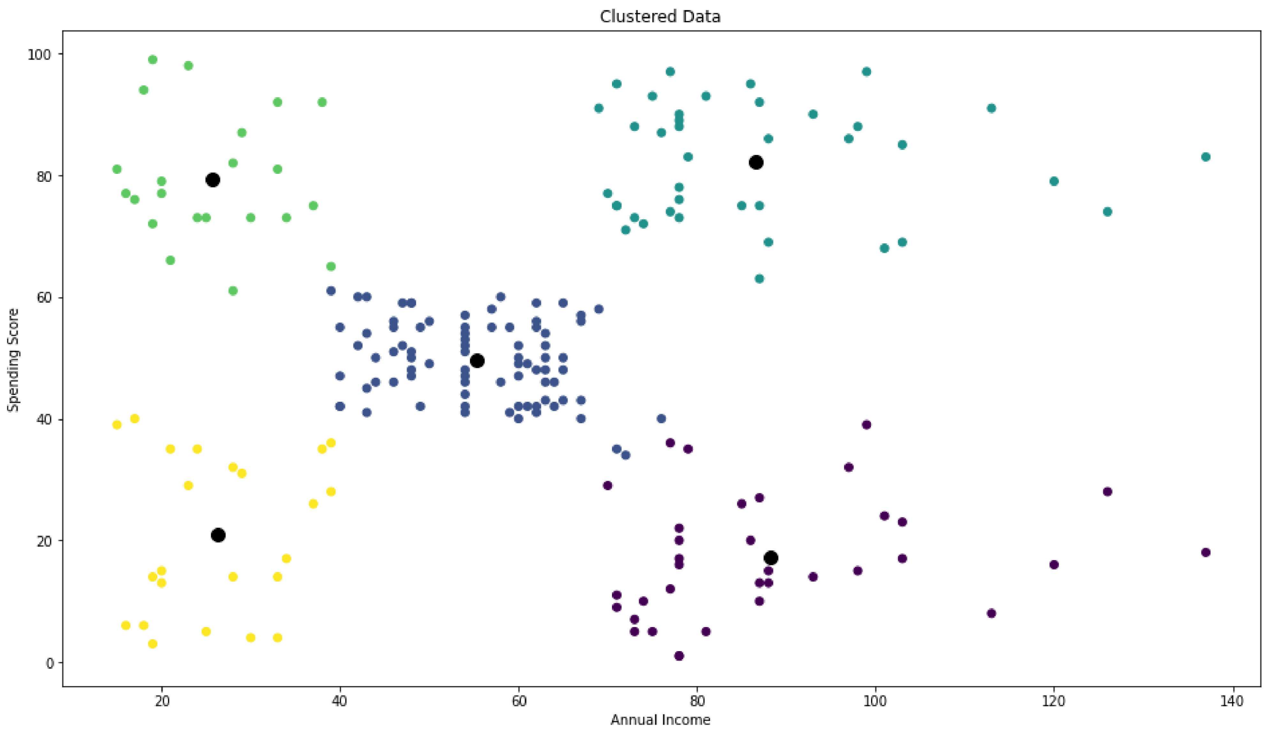
Out[23]:
```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

In [25]:
```
#calculate centroids
cent=km.cluster_centers_
```

In [27]:
```
#plot using kmeans
plt.figure(figsize=(16,9))
plt.title('Clustered Data')
```

```
plt.xlabel('Annual 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[27]: <matplotlib.collections.PathCollection at 0x241097e8fa0>

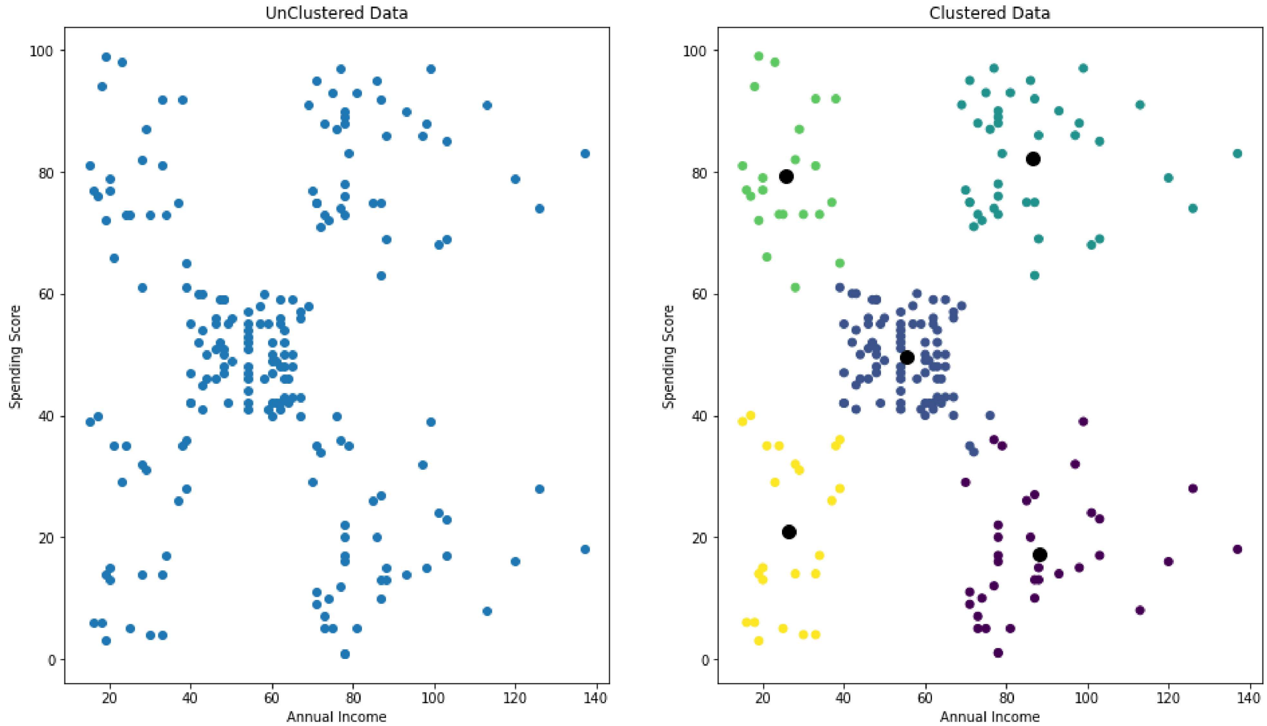


In [31]:
```
#Compare unclustered and clustered

plt.figure(figsize=(16,9))
plt.subplot(1,2,1)
plt.title('UnClustered Data')
plt.xlabel('Annual 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('Annual 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[31]: <matplotlib.collections.PathCollection at 0x24109abbcd0>

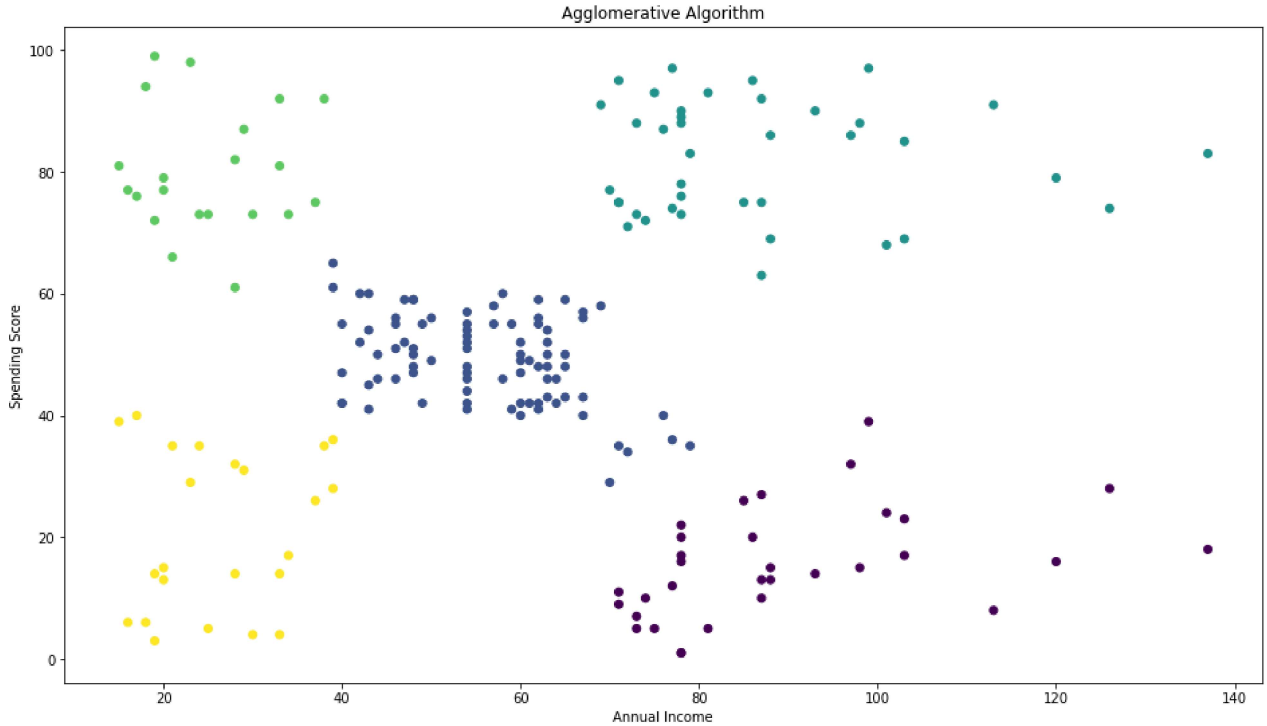


In [34]:
```
#using agglomerative Algorithm
agl=AgglomerativeClustering(n_clusters=5)
alabels=agl.fit_predict(x)
alabels
```

Out[34]:
```
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, 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, 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], dtype=int64)
```

In [35]:
```
plt.figure(figsize=(16,9))
plt.title('Agglomerative Algorithm')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x[ 'Spending Score (1-100)'],c=alabels)
```

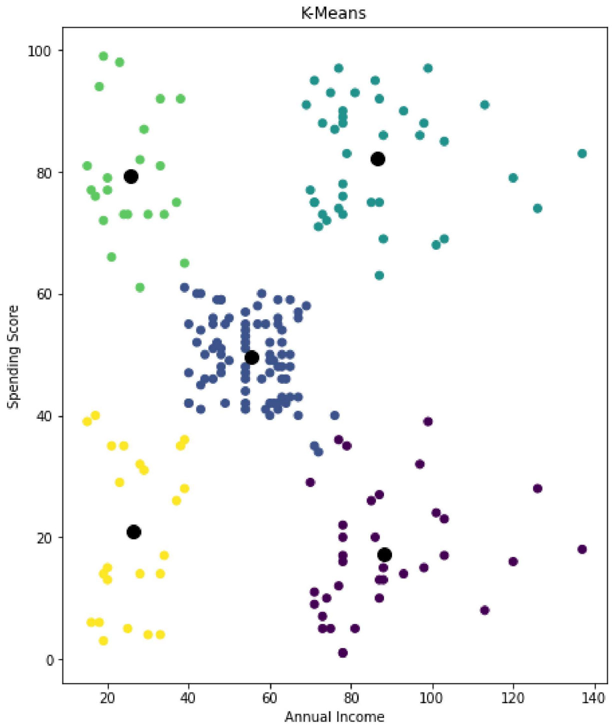
Out[35]:
```
<matplotlib.collections.PathCollection at 0x24109bcd040>
```

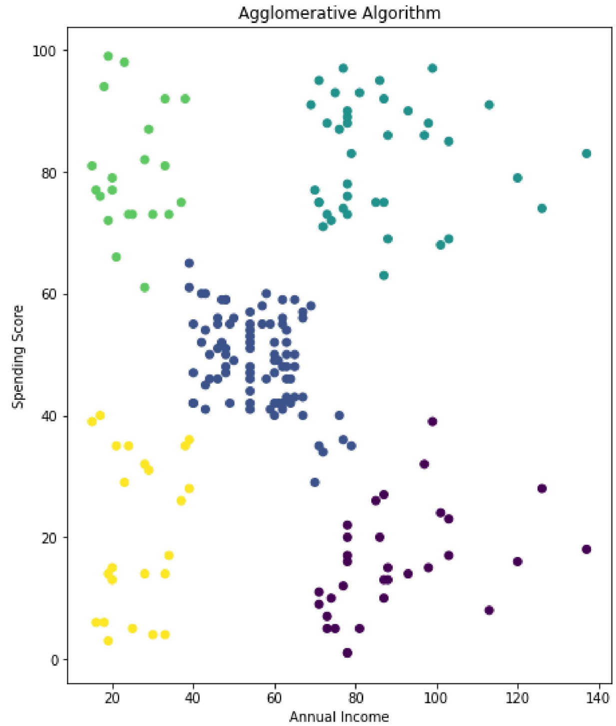


In [36]:
```
#compare 2 clustering algorithms
plt.figure(figsize=(16,9))
plt.subplot(1,2,1)
plt.title('K-Means')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(x['Annual Income (k$)'],x[ 'Spending Score (1-100)'],c=labels)
plt.scatter(cent[:,0], cent[:,1],s=100,color='k')

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

Out[36]: <matplotlib.collections.PathCollection at 0x24109c5c7c0>


K-Means
Spending Score
Annual Income
0
20
40
60
80
100
20
40
60
80
100
120
140


Agglomerative Algorithm
Spending Score
Annual Income
0
20
40
60
80
100
20
40
60
80
100
120
140

In [ ]: