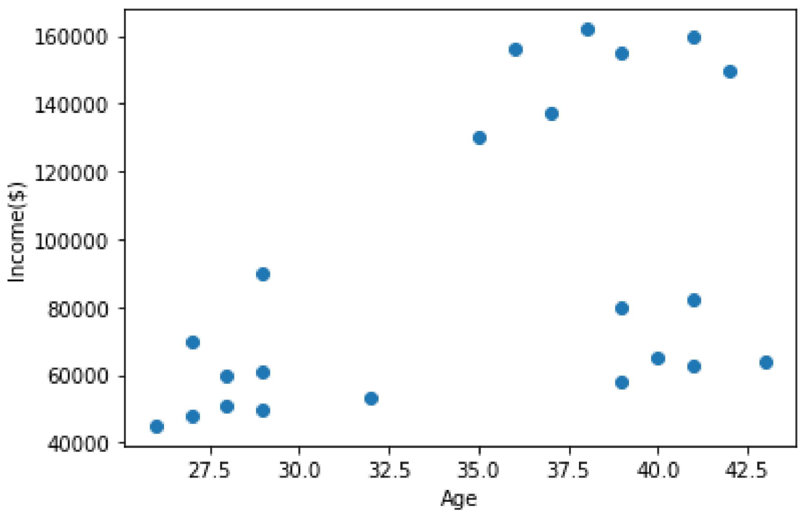
In [1]:
```
#In this module,  we talked about cluster analysis. In our hierarchical  algorith
import pandas as pd
import sklearn as sk
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import AgglomerativeClustering

df = pd.read_csv('OneDrive\Desktop\income.csv')

plt.scatter(df.Age,df['Income($)'])
plt.xlabel('Age')
plt.ylabel('Income($)')
```

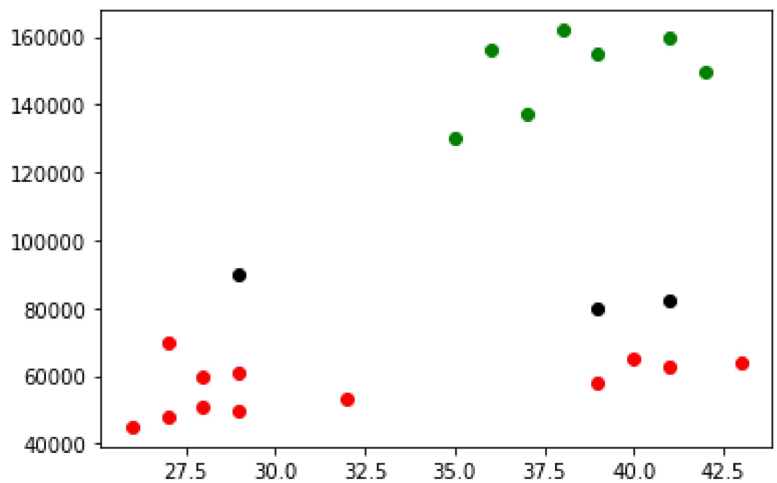
Out[1]:
```
Text(0, 0.5, 'Income($)')
```



In [4]:
```
ac = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
df['cluster']=ac.fit_predict(df[['Age','Income($)']])
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')

df.head()
```

Out[4]:

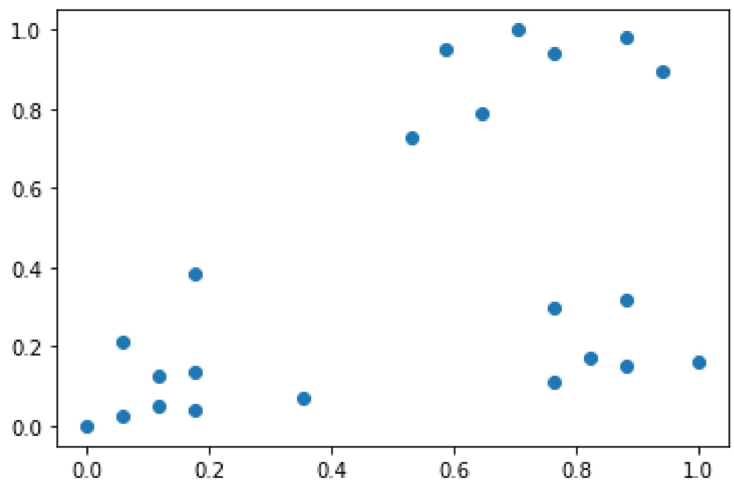
	Name	Age	Income($)	cluster
0	Rob	27	70000	1
1	Michael	29	90000	2
2	Mohan	29	61000	1
3	Ismail	28	60000	1
4	Kory	42	150000	0



In [7]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(df[['Income($)']])
df['Income($)'] = scaler.transform(df[['Income($)']])
scaler.fit(df[['Age']])
df['Age'] = scaler.transform(df[['Age']])
plt.scatter(df.Age,df['Income($)'])

df.head()
```

Out[7]:

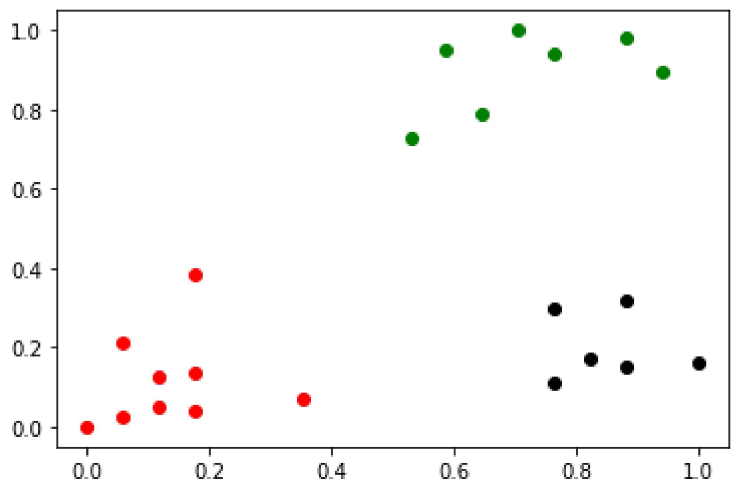
	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	1
1	Michael	0.176471	0.384615	1
2	Mohan	0.176471	0.136752	1
3	Ismail	0.117647	0.128205	1
4	Kory	0.941176	0.897436	0



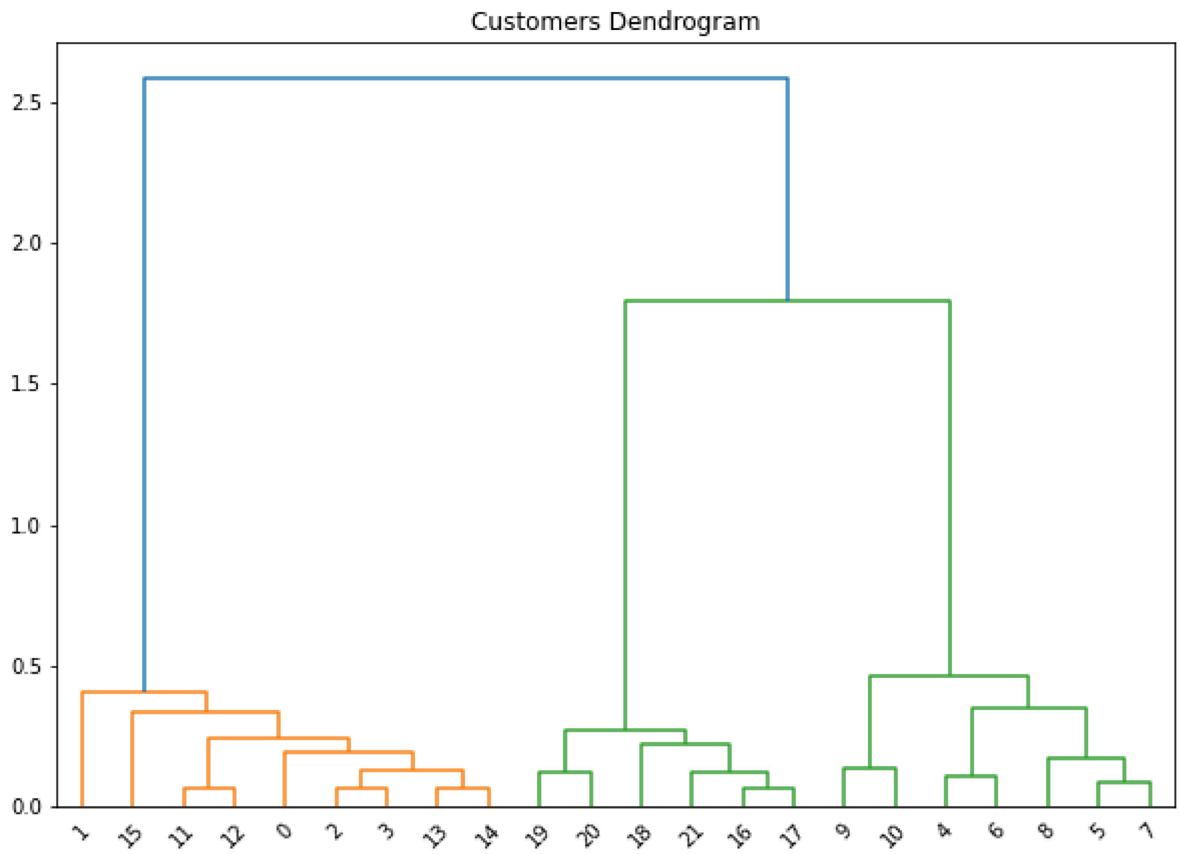
In [9]:
```
acs = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
df['cluster']=acs.fit_predict(df[['Age','Income($)']])
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
plt.scatter(df3.Age,df3['Income($)'],color='black')

df.head()
```

Out[9]:

	Name	Age	Income($)	cluster
0	Rob	0.058824	0.213675	1
1	Michael	0.176471	0.384615	1
2	Mohan	0.176471	0.136752	1
3	Ismail	0.117647	0.128205	1
4	Kory	0.941176	0.897436	0

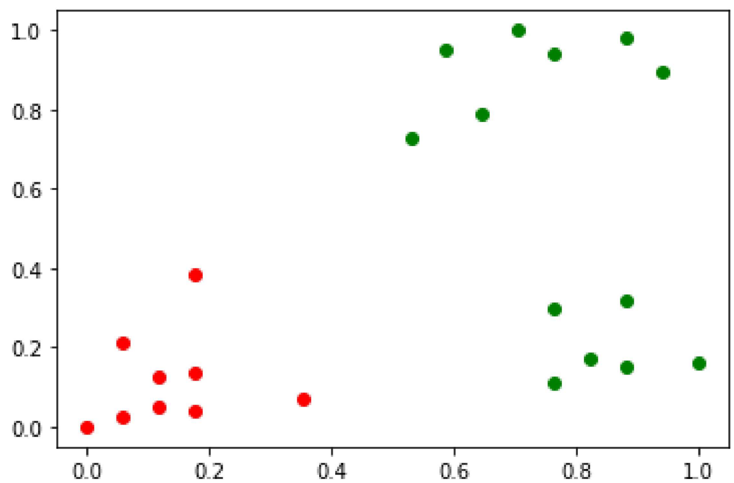


In [34]:
```
# 1. Can you build a new model?  For your new model, you can change number of clu

#2. Please compare your new model with our old one, and summarize your result.

#https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/
import scipy.cluster.hierarchy as shc

plt.figure(figsize=(10, 7))
plt.title("Customers Dendrogram")

# Selecting Annual Income and Spending Scores by index
selected_data = df.iloc[:, 1:3]
selected_data.head()

#using dendogram to find the ideal number of clusters using ward and euclidean
clusters = shc.linkage(selected_data,
            method='ward',
            metric="euclidean")
shc.dendrogram(Z=clusters)
plt.show()
```



In [38]:
```
acs2 = AgglomerativeClustering(n_clusters=2, affinity='euclidean', linkage='ward'
df['cluster']=acs2.fit_predict(df[['Age','Income($)']])
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]

plt.scatter(df1.Age,df1['Income($)'],color='green')
plt.scatter(df2.Age,df2['Income($)'],color='red')
```

Out[38]:
```
<matplotlib.collections.PathCollection at 0x2a88c50b130>
```



In [40]:
```
#Going down to 2 clusters, it appears age is the determining factor.
#I believe the 3 cluster output was more accurate.

#Much more work can be done on this discussion. Increasing the clustering nodes m
#We can also change the distance measurement method and linkage.
#A more dynamic clustering method would be interesting to implement but a "pre-co
#distance matrix prior to implementation.
```

In [ ]:
```
```