import libraries

In [1]:
```
import numpy as np
```

In [2]:
```
import pandas as pd
country_data= pd.read_csv('Country-data.csv')
```

In [3]:
```
country_data.shape
```

Out[3]:
```
(167, 10)
```

In [4]:
```
country_data.head
```

Out[4]:
```
<bound method NDFrame.head of                 country  child_mort  exports  he
alth  imports  income  \
0           Afghanistan        90.2     10.0    7.58     44.9    1610
1               Albania        16.6     28.0    6.55     48.6    9930
2               Algeria        27.3     38.4    4.17     31.4   12900
3                Angola       119.0     62.3    2.85     42.9    5900
4    Antigua and Barbuda        10.3     45.5    6.03     58.9   19100
..                   ...         ...      ...     ...      ...     ...
162             Vanuatu        29.2     46.6    5.25     52.7    2950
163           Venezuela        17.1     28.5    4.91     17.6   16500
164             Vietnam        23.3     72.0    6.84     80.2    4490
165               Yemen        56.3     30.0    5.18     34.4    4480
166              Zambia        83.1     37.0    5.89     30.9    3280

     inflation  life_expec  total_fer   gdpp
0         9.44        56.2       5.82    553
1         4.49        76.3       1.65   4090
2        16.10        76.5       2.89   4460
3        22.40        60.1       6.16   3530
4         1.44        76.8       2.13  12200
..         ...         ...        ...    ...
162       2.62        63.0       3.50   2970
163      45.90        75.4       2.47  13500
164      12.10        73.1       1.95   1310
165      23.60        67.5       4.67   1310
166      14.00        52.0       5.40   1460

[167 rows x 10 columns]>
```

drop duplicated rows

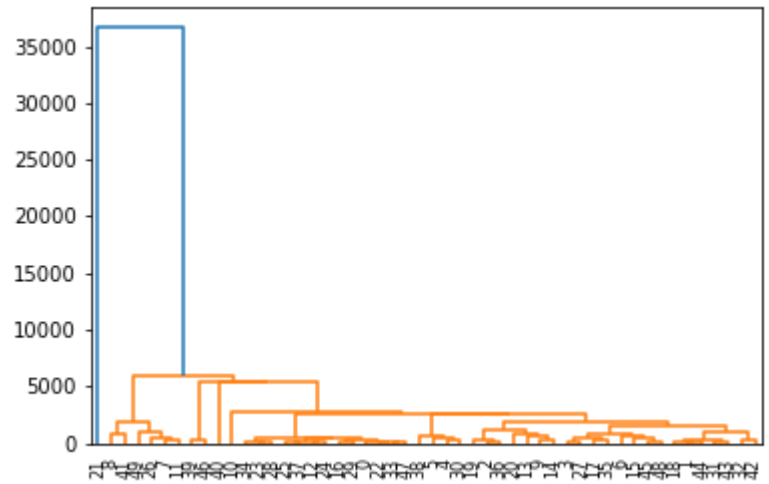
In [5]:
```
country_data = country_data.drop_duplicates(subset='health', keep="first")
country_data = country_data.drop_duplicates(subset='income', keep="first")
country_data.head
```

Out[5]:
```
<bound method NDFrame.head of                 country  child_mort  exports  he
alth  imports  income  \
0           Afghanistan        90.2     10.0    7.58     44.9    1610
1               Albania        16.6     28.0    6.55     48.6    9930
2               Algeria        27.3     38.4    4.17     31.4   12900
3                Angola       119.0     62.3    2.85     42.9    5900
4    Antigua and Barbuda        10.3     45.5    6.03     58.9   19100
..                   ...         ...      ...     ...      ...     ...
161           Uzbekistan        36.3     31.7    5.81     28.5    4240
163            Venezuela        17.1     28.5    4.91     17.6   16500
164              Vietnam        23.3     72.0    6.84     80.2    4490
165                Yemen        56.3     30.0    5.18     34.4    4480
166               Zambia        83.1     37.0    5.89     30.9    3280

     inflation  life_expec  total_fer   gdpp
0         9.44        56.2       5.82    553
1         4.49        76.3       1.65   4090
2        16.10        76.5       2.89   4460
3        22.40        60.1       6.16   3530
4         1.44        76.8       2.13  12200
..         ...         ...        ...    ...
161      16.50        68.8       2.34   1380
163      45.90        75.4       2.47  13500
164      12.10        73.1       1.95   1310
165      23.60        67.5       4.67   1310
166      14.00        52.0       5.40   1460

[137 rows x 10 columns]>
```

needed features

In [6]:
```
data = country_data.iloc[:, [3,5]].values
```

In [7]:
```
import scipy.cluster.hierarchy as sch
dendrogram= sch.dendrogram(sch.linkage(data[:50],"single"))
```



object from hierarchical class#

In [8]:
```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean',linkage='sir
cluster.fit_predict(data)
```

Out[8]:
```
array([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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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], dtype=int64)
```

scattering data

In [9]:
```
import matplotlib.pyplot as plt
plt.scatter (data[:,0],data[:,1], c=cluster.labels_, cmap='rainbow')
plt.title('cluster of country')
plt.xlabel('Income($)')
plt.ylabel('Health care (1-167)')
```

Out[9]:
```
Text(0, 0.5, 'Health care (1-167)')
```