In [588]:
```
#imports

import pandas as pd
import scipy.cluster.hierarchy as sch
from sklearn import preprocessing
from sklearn.cluster import AgglomerativeClustering
from sklearn_extra.cluster import KMedoids
import matplotlib.pyplot as plt
```

In [589]:
```
#read data file

happiness_data= pd.read_csv('world-happiness-report-2021.csv')
```

In [590]:
```
#shape of data

happiness_data.shape
```

Out[590]: (149, 20)

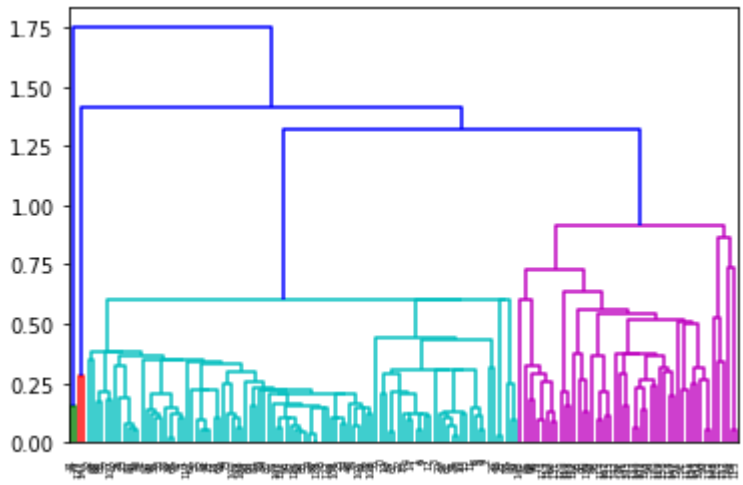
In [591]:
```
# describe data

happiness_data.describe()
```

Out[591]:

	Ladder score	Standard error of ladder score	upperwhisker	lowerwhisker	Logged GDP per capita	Social support	Healthy life expectancy
count	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000	149.000000
mean	5.532839	0.058752	5.648007	5.417631	9.432208	0.814745	64.992799
std	1.073924	0.022001	1.054330	1.094879	1.158601	0.114889	6.762043
min	2.523000	0.026000	2.596000	2.449000	6.635000	0.463000	48.478000
25%	4.852000	0.043000	4.991000	4.706000	8.541000	0.750000	59.802000
50%	5.534000	0.054000	5.625000	5.413000	9.569000	0.832000	66.603000
75%	6.255000	0.070000	6.344000	6.128000	10.421000	0.905000	69.600000
max	7.842000	0.173000	7.904000	7.780000	11.647000	0.983000	76.953000

In [592]:
```
# needed features

data1 = happiness_data.iloc[:,[8,10]].values
```

In [593]:
```
#shows the hierarchical relationship between objects.

dendrogram= sch.dendrogram(sch.linkage(data1,'single'))
```

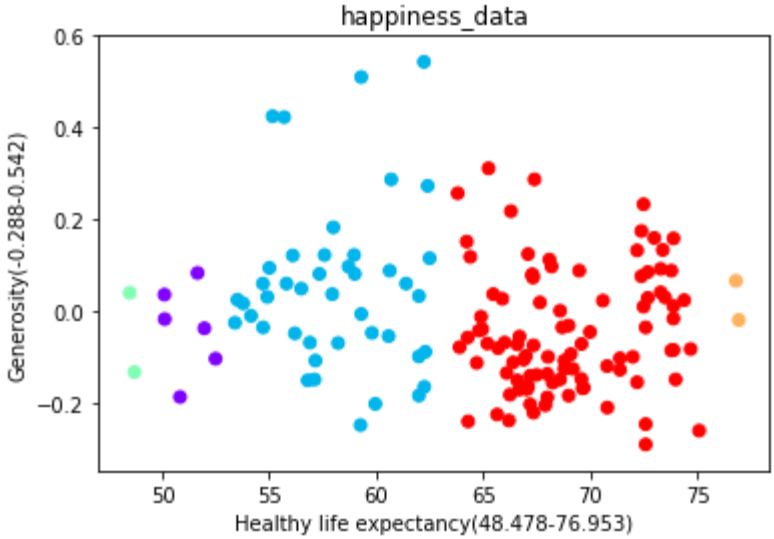


In [594]:
```
#create object from Hierarchial class

cluster = AgglomerativeClustering(n_clusters=5,affinity='euclidean', lir
cluster.fit_predict(data1)
```

Out[594]:
```
array([4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 4, 4, 4, 4,
4,
       4, 4, 4, 1, 4, 4, 4, 4, 4, 4, 3, 4, 4, 1, 4, 1, 1, 4, 0, 4, 4,
4,
       4, 4, 1, 1, 4, 4, 1, 1, 1, 1, 1, 1, 4, 1, 1, 4, 1, 4, 4, 4, 4,
4,
       1, 1, 1, 1, 1, 0, 0, 4, 1, 1, 1, 4, 4, 1, 1, 1, 4, 2, 4, 0, 1,
1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 0], dtype=int6
4)
```

In [595]:
```
#scattering data

plt.scatter(data1[:,0], data1[:,1], c=cluster.labels_, cmap='rainbow')
plt.title('happiness_data')
plt.xlabel('Healthy life expectancy(48.478-76.953)')
plt.ylabel('Generosity(-0.288-0.542)')
```

Out[595]:
```
Text(0,0.5,'Generosity(-0.288-0.542)')
```



In [596]:
```
# needed features

data2 = happiness_data.iloc[:, [8,10]].values
```

In [597]:
```
#create object from KMedoids class

cluster = KMedoids(n_clusters=3, metric="manhattan",init="random",randon
cluster.fit_predict(data2)
```

Out[597]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
1,
       0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0,
1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1,
1,
       1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 2, 1, 2, 1, 1,
1,
       0, 1, 2, 2, 1, 1, 2, 2, 1, 2, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 1,
1,
       2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2,
1,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2], dtype=int6
4)
```

In [598]:
```
#scattering data

plt.scatter(data2[:,0], data2[:,1], c=cluster.labels_, cmap='rainbow')
plt.title('happiness_data')
plt.xlabel('Healthy life expectancy(48.478-76.953)')
plt.ylabel('Generosity(-0.288-0.542))')
```

Out[598]:
```
Text(0,0.5,'Generosity(-0.288-0.542))')
```