Hierarchical-Clustering

In [1]:

```
# import hierarchical clustering libraries
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [2]:

```
univ=pd.read_csv("C:/Users/Ashraf/Documents/Datafiles/Universities.csv")
```

In [3]:

```
univ.head()
```

Out[3]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	Brown	1310	89	22	13	22704	94
1	CalTech	1415	100	25	6	63575	81
2	CMU	1260	62	59	9	25026	72
3	Columbia	1310	76	24	12	31510	88
4	Cornell	1280	83	33	13	21864	90

In [4]:

```
univ.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Univ      25 non-null     object
 1   SAT       25 non-null     int64 
 2   Top10     25 non-null     int64 
 3   Accept    25 non-null     int64 
 4   SFRatio   25 non-null     int64 
 5   Expenses  25 non-null     int64 
 6   GradRate  25 non-null     int64 
dtypes: int64(6), object(1)
memory usage: 1.5+ KB
```

In [5]:

```
# Normalization Function
def norm_func(i):
    x=(i-i.min())/(i.max()-i.min())
    return (x)
```

In [6]:

```
# Normalized dataframe
df_norm=norm_func(univ.iloc[:,1:])
```

In [7]:

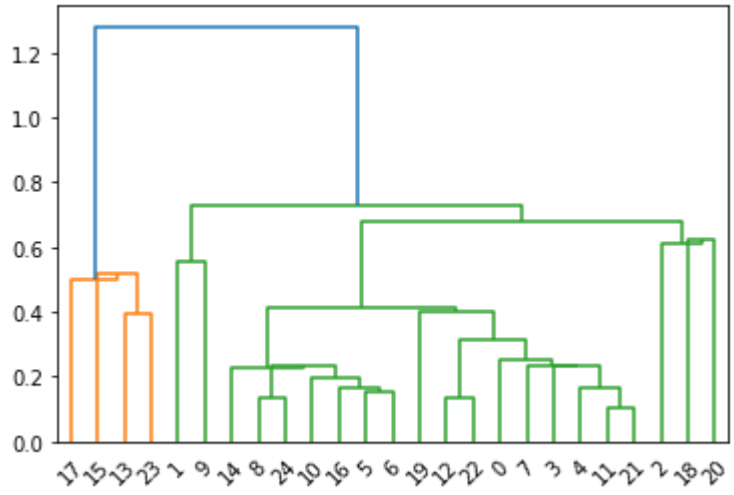
```
df_norm
```

Out[7]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	0.743902	0.847222	0.105263	0.368421	0.255144	0.900000
1	1.000000	1.000000	0.144737	0.000000	1.000000	0.466667
2	0.621951	0.472222	0.592105	0.157895	0.297461	0.166667
3	0.743902	0.666667	0.131579	0.315789	0.415629	0.700000
4	0.670732	0.763889	0.250000	0.368421	0.239835	0.766667
5	0.817073	0.847222	0.118421	0.210526	0.427512	0.933333
6	0.756098	0.861111	0.210526	0.315789	0.416996	0.933333
7	0.609756	0.638889	0.131579	0.315789	0.208161	0.833333
8	0.963415	0.875000	0.000000	0.263158	0.561699	1.000000
9	0.731707	0.652778	0.394737	0.052632	0.910991	0.666667
10	0.914634	0.916667	0.210526	0.210526	0.476864	0.800000
11	0.621951	0.791667	0.328947	0.263158	0.352609	0.733333
12	0.609756	0.736111	0.368421	0.368421	0.116965	0.900000
13	0.185366	0.138889	0.526316	0.631579	0.026991	0.433333
14	0.902439	0.875000	0.000000	0.105263	0.392120	0.933333
15	0.000000	0.000000	1.000000	0.684211	0.006597	0.066667
16	0.865854	0.861111	0.078947	0.315789	0.505659	0.866667
17	0.170732	0.291667	0.697368	1.000000	0.000000	0.000000
18	0.573171	0.930556	0.342105	0.578947	0.117293	0.366667
19	0.695122	0.652778	0.473684	0.368421	0.540832	0.666667
20	0.426829	0.513889	0.710526	0.526316	0.123307	0.600000
21	0.682927	0.722222	0.289474	0.263158	0.343515	0.766667
22	0.536585	0.680556	0.394737	0.421053	0.084653	0.833333
23	0.195122	0.166667	0.723684	0.473684	0.057462	0.133333
24	0.902439	0.930556	0.065789	0.263158	0.634397	0.966667

In [8]:

```
# create dendrogram
dendrogram=sch.dendrogram(sch.linkage(df_norm, method='centroid'))
```



In [9]:

```
# Create clusters
hc=AgglomerativeClustering(n_clusters=4, affinity='euclidean',linkage='ward')
```

In [10]:

```
hc
```

Out[10]:

```
AgglomerativeClustering(n_clusters=4)
```

In [11]:

```
# save cluster for chart
y_hc=hc.fit_predict(df_norm)
Clusters=pd.DataFrame(y_hc, columns=['Clusters'])
```

In [12]:

```
Clusters
```

Out[12]:

	Clusters
0	3
1	0
2	2
3	3
4	3
5	0
6	0
7	3
8	0
9	0
10	0
11	3
12	3
13	1
14	0
15	1
16	0
17	1
18	2
19	0
20	2
21	3
22	3
23	1
24	0

In [13]:

```
df_norm['h_clusterid'] = hc.labels_
```

In [18]:

```
univ['h_clusterid']=hc.labels_
```

In [19]:

```
univ.groupby('h_clusterid').agg(['mean']).reset_index()
```

Out[19]:

	h_clusterid	SAT	Top10	Accept	SFRatio	Expenses	GradRate
		mean	mean	mean	mean	mean	mean
0	0	1355.500000	89.000	26.900000	10.000	40897.200000	91.700000
1	1	1061.500000	38.750	70.000000	19.250	9953.000000	71.750000
2	2	1226.666667	74.000	55.666667	14.000	18545.333333	78.333333
3	3	1272.500000	80.625	33.000000	12.375	22535.000000	91.125000

In [20]:

```
univ
```

Out[20]:

	Univ	SAT	Top10	Accept	SFRatio	Expenses	GradRate	h_clusterid
0	Brown	1310	89	22	13	22704	94	3
1	CalTech	1415	100	25	6	63575	81	0
2	CMU	1260	62	59	9	25026	72	2
3	Columbia	1310	76	24	12	31510	88	3
4	Cornell	1280	83	33	13	21864	90	3
5	Dartmouth	1340	89	23	10	32162	95	0
6	Duke	1315	90	30	12	31585	95	0
7	Georgetown	1255	74	24	12	20126	92	3
8	Harvard	1400	91	14	11	39525	97	0
9	JohnsHopkins	1305	75	44	7	58691	87	0
10	MIT	1380	94	30	10	34870	91	0
11	Northwestern	1260	85	39	11	28052	89	3
12	NotreDame	1255	81	42	13	15122	94	3
13	PennState	1081	38	54	18	10185	80	1
14	Princeton	1375	91	14	8	30220	95	0
15	Purdue	1005	28	90	19	9066	69	1
16	Stanford	1360	90	20	12	36450	93	0
17	TexasA&M	1075	49	67	25	8704	67	1
18	UCBerkeley	1240	95	40	17	15140	78	2
19	UChicago	1290	75	50	13	38380	87	0
20	UMichigan	1180	65	68	16	15470	85	2
21	UPenn	1285	80	36	11	27553	90	3
22	UVA	1225	77	44	14	13349	92	3
23	UWisconsin	1085	40	69	15	11857	71	1
24	Yale	1375	95	19	11	43514	96	0

In [21]:

```
hc.labels_
```

Out[21]:

```
array([3, 0, 2, 3, 3, 0, 0, 3, 0, 0, 0, 3, 3, 1, 0, 1, 0, 1, 2, 0, 2, 3,
       3, 1, 0], dtype=int64)
```

In [22]:

```
df_norm
```

Out[22]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate	h_clusterid
0	0.743902	0.847222	0.105263	0.368421	0.255144	0.900000	3
1	1.000000	1.000000	0.144737	0.000000	1.000000	0.466667	0
2	0.621951	0.472222	0.592105	0.157895	0.297461	0.166667	2
3	0.743902	0.666667	0.131579	0.315789	0.415629	0.700000	3
4	0.670732	0.763889	0.250000	0.368421	0.239835	0.766667	3
5	0.817073	0.847222	0.118421	0.210526	0.427512	0.933333	0
6	0.756098	0.861111	0.210526	0.315789	0.416996	0.933333	0
7	0.609756	0.638889	0.131579	0.315789	0.208161	0.833333	3
8	0.963415	0.875000	0.000000	0.263158	0.561699	1.000000	0
9	0.731707	0.652778	0.394737	0.052632	0.910991	0.666667	0
10	0.914634	0.916667	0.210526	0.210526	0.476864	0.800000	0
11	0.621951	0.791667	0.328947	0.263158	0.352609	0.733333	3
12	0.609756	0.736111	0.368421	0.368421	0.116965	0.900000	3
13	0.185366	0.138889	0.526316	0.631579	0.026991	0.433333	1
14	0.902439	0.875000	0.000000	0.105263	0.392120	0.933333	0
15	0.000000	0.000000	1.000000	0.684211	0.006597	0.066667	1
16	0.865854	0.861111	0.078947	0.315789	0.505659	0.866667	0
17	0.170732	0.291667	0.697368	1.000000	0.000000	0.000000	1
18	0.573171	0.930556	0.342105	0.578947	0.117293	0.366667	2
19	0.695122	0.652778	0.473684	0.368421	0.540832	0.666667	0
20	0.426829	0.513889	0.710526	0.526316	0.123307	0.600000	2
21	0.682927	0.722222	0.289474	0.263158	0.343515	0.766667	3
22	0.536585	0.680556	0.394737	0.421053	0.084653	0.833333	3
23	0.195122	0.166667	0.723684	0.473684	0.057462	0.133333	1
24	0.902439	0.930556	0.065789	0.263158	0.634397	0.966667	0

In [ ]:

```

```