In [1]:
```
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('Universities.csv')
df.head()
```

Out[2]:

	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 [3]:
```
df.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]:
```
def normfunc(i):
    x = (i-i.min())/(i.max()-i.min())
    return x
```

In [7]:
```
df_norm = normfunc(df.iloc[:,1:])
```

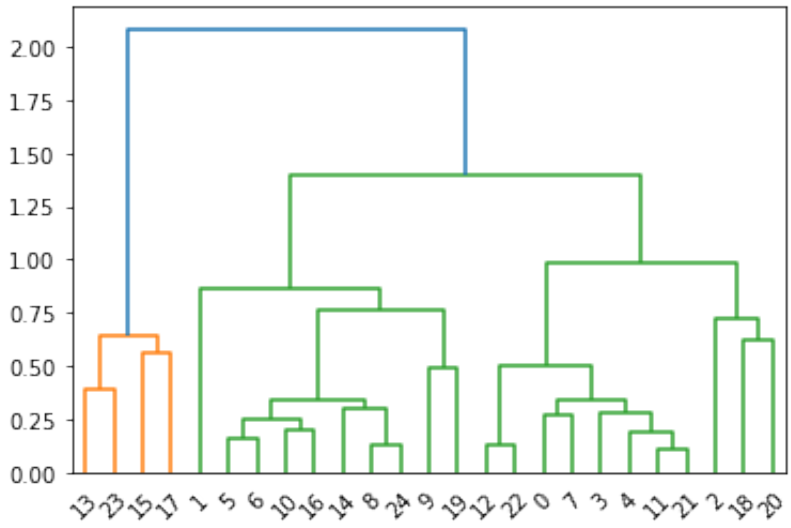
Type *Markdown* and LaTeX: α^2

In [8]:
```
df_norm
```

Out[8]:

	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 [9]:
```
dendrogram = sch.dendrogram(sch.linkage(df_norm, method = "complete
```



In [10]:
```
hc = AgglomerativeClustering(n_clusters=5, affinity='euclidean', li
```

In [11]:
```
hc.fit(df_norm)
```

Out[11]:
```
AgglomerativeClustering(linkage='complete', n_clusters=5)
```

In [12]:
```
y_hc = hc.fit_predict(df_norm)
```

In [13]:
```
y_hc
```

Out[13]:
```
array([3, 4, 2, 3, 3, 0, 0, 3, 0, 0, 0, 3, 3, 1, 0, 1, 0, 1, 2, 0,
2, 3,
       3, 1, 0])
```

In [14]:
```
df['h_clusterid']= y_hc
```

In [15]:
```
df.head()
```

Out[15]:

	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	4
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

In [16]:
```
df1 = df.sort_values('h_clusterid')
df1.iloc[:,[0,-1]]
```

Out[16]:

	Univ	h_clusterid
24	Yale	0
14	Princeton	0
10	MIT	0
9	JohnsHopkins	0
16	Stanford	0
19	UChicago	0
8	Harvard	0
5	Dartmouth	0
6	Duke	0
23	UWisconsin	1
13	PennState	1
15	Purdue	1
17	TexasA&M	1
20	UMichigan	2
2	CMU	2
18	UCBerkeley	2
21	UPenn	3
22	UVA	3
0	Brown	3
7	Georgetown	3
4	Cornell	3
3	Columbia	3
11	Northwestern	3
12	NotreDame	3
1	CalTech	4

In [ ]: