In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df=pd.read_csv("C:/Users/USER/Desktop/Datasets/University_Clustering.csv")
```

In [3]:

```
del df["Univ"]
del df["State"]
df.head(7)
```

Out[3]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate
0	1310	89	22	13	22,704	94
1	1415	100	25	6	63,575	81
2	1260	62	59	9	25,026	72
3	1310	76	24	12	31,510	88
4	1280	83	33	13	21,864	90
5	1340	89	23	10	32,162	95
6	1315	90	30	12	31,585	95

In [4]:

```
df['Expenses']=df['Expenses'].str.replace(',','').astype(int)
```

In [5]:

```
df.describe()
```

Out[5]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate
count	25.000000	25.000000	25.000000	25.00000	25.000000	25.000000
mean	1266.440000	76.480000	39.200000	12.72000	27388.000000	86.720000
std	108.359771	19.433905	19.727308	4.06735	14424.883165	9.057778
min	1005.000000	28.000000	14.000000	6.00000	8704.000000	67.000000
25%	1240.000000	74.000000	24.000000	11.00000	15140.000000	81.000000
50%	1285.000000	81.000000	36.000000	12.00000	27553.000000	90.000000
75%	1340.000000	90.000000	50.000000	14.00000	34870.000000	94.000000
max	1415.000000	100.000000	90.000000	25.00000	63575.000000	97.000000

In [6]:

```
from sklearn.preprocessing import scale
df_n = scale(df)
df_n.shape
```

Out[6]:

```
(25, 6)
```

In [7]:

```
from sklearn.decomposition import PCA
pca = PCA(n_components=4)
pca_values = pca.fit_transform(df_n)
pca_values
```

Out[7]:

```
array([[-1.00987445, -1.06430962,  0.08106631,  0.05695064],
       [-2.82223781,  2.25904458,  0.83682883,  0.14384464],
       [ 1.11246577,  1.63120889, -0.26678684,  1.07507502],
       [-0.74174122, -0.04218747,  0.06050086, -0.15720812],
       [-0.31191206, -0.63524357,  0.01024052,  0.17136367],
       [-1.69669089, -0.34436328, -0.25340751,  0.01256433],
       [-1.24682093, -0.49098366, -0.03209382, -0.20564378],
       [-0.33874978, -0.78516859, -0.49358483,  0.03985631],
       [-2.37415013, -0.38653888,  0.11609839, -0.45336562],
       [-1.40327739,  2.11951503, -0.44282714, -0.63254327],
       [-1.72610332,  0.08823712,  0.17040366,  0.26090191],
       [-0.45085748, -0.01113295, -0.17574605,  0.23616563],
       [ 0.04023814, -1.00920438, -0.49651717,  0.22929876],
       [ 3.23373034, -0.37458049, -0.49537282, -0.52123771],
       [-2.23626502, -0.37179329, -0.39899365,  0.40696648],
       [ 5.17299212,  0.77991535, -0.38591233, -0.23221171],
       [-1.69964377, -0.30559745,  0.31850785, -0.29746268],
       [ 4.578146  , -0.34759136,  1.49964176, -0.45425171],
       [ 0.82260312, -0.69890615,  1.42781145,  0.7607788 ],
       [-0.09776213,  0.65044645,  0.10050844, -0.50009719],
       [ 1.9631826 , -0.22476756, -0.25588143, -0.0484741 ],
       [-0.54228894, -0.07958884, -0.30539348,  0.13169876],
       [ 0.53222092, -1.0171672 , -0.42371636,  0.16953571],
       [ 3.54869664,  0.77846167, -0.44936332,  0.32367862],
       [-2.30590032, -0.11770432,  0.25398866, -0.51618337]])
```

In [8]:

```
names = df.columns
names
```

Out[8]:

```
Index(['SAT', 'Top10', 'Accept', 'SFRatio', 'Expenses', 'GradRate'], dtype='obj
ect')
```

In [9]:
```
pdf = pd.DataFrame(pca_values)
pdf
```

Out[9]:

	0	1	2	3
0	-1.009874	-1.064310	0.081066	0.056951
1	-2.822238	2.259045	0.836829	0.143845
2	1.112466	1.631209	-0.266787	1.075075
3	-0.741741	-0.042187	0.060501	-0.157208
4	-0.311912	-0.635244	0.010241	0.171364
5	-1.696691	-0.344363	-0.253408	0.012564
6	-1.246821	-0.490984	-0.032094	-0.205644
7	-0.338750	-0.785169	-0.493585	0.039856
8	-2.374150	-0.386539	0.116098	-0.453366
9	-1.403277	2.119515	-0.442827	-0.632543
10	-1.726103	0.088237	0.170404	0.260902
11	-0.450857	-0.011133	-0.175746	0.236166
12	0.040238	-1.009204	-0.496517	0.229299
13	3.233730	-0.374580	-0.495373	-0.521238
14	-2.236265	-0.371793	-0.398994	0.406966
15	5.172992	0.779915	-0.385912	-0.232212
16	-1.699644	-0.305597	0.318508	-0.297463
17	4.578146	-0.347591	1.499642	-0.454252
18	0.822603	-0.698906	1.427811	0.760779
19	-0.097762	0.650446	0.100508	-0.500097
20	1.963183	-0.224768	-0.255881	-0.048474
21	-0.542289	-0.079589	-0.305393	0.131699
22	0.532221	-1.017167	-0.423716	0.169536
23	3.548697	0.778462	-0.449363	0.323679
24	-2.305900	-0.117704	0.253989	-0.516183

In [10]:
```
var = pca.explained_variance_ratio_
var
```

Out[10]:
```
array([0.76868084, 0.13113602, 0.04776031, 0.02729668])
```

In [11]:
```
v_1 = np.cumsum(np.round(var,decimals = 2)*100)
v_1
```

Out[11]:
```
array([77., 90., 95., 98.])
```

In [14]:

```
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
mdl=KMeans(n_clusters=3)
end = mdl.fit(pdf)
end
```

Out[14]:

```
KMeans(n_clusters=3)
```

In [16]:

```
y_kmeans = mdl.predict(pdf)
y_kmeans
```

Out[16]:

```
array([2, 0, 2, 2, 2, 0, 0, 2, 0, 0, 0, 2, 2, 1, 0, 1, 0, 1, 2, 2, 1, 2,
       2, 1, 0])
```

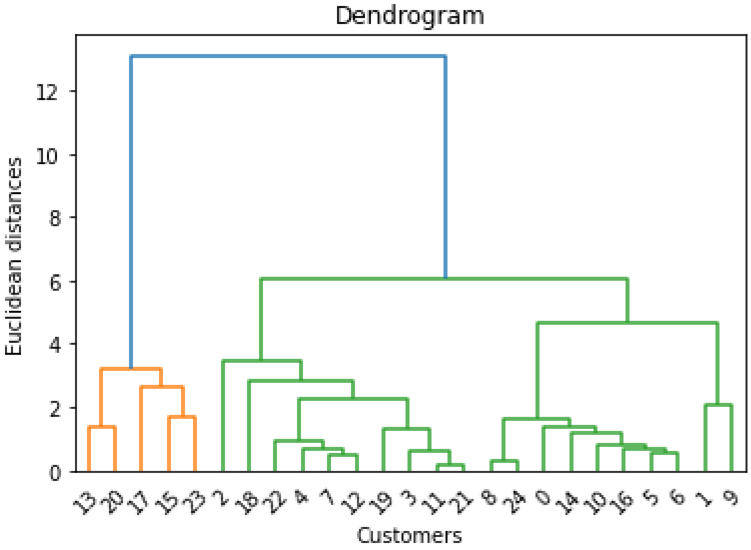
In [17]:

```
df['clusters']=pd.Series(y_kmeans)
df
```

Out[17]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate	clusters
0	1310	89	22	13	22704	94	2
1	1415	100	25	6	63575	81	0
2	1260	62	59	9	25026	72	2
3	1310	76	24	12	31510	88	2
4	1280	83	33	13	21864	90	2
5	1340	89	23	10	32162	95	0
6	1315	90	30	12	31585	95	0
7	1255	74	24	12	20126	92	2
8	1400	91	14	11	39525	97	0
9	1305	75	44	7	58691	87	0
10	1380	94	30	10	34870	91	0
11	1260	85	39	11	28052	89	2
12	1255	81	42	13	15122	94	2
13	1081	38	54	18	10185	80	1
14	1375	91	14	8	30220	95	0
15	1005	28	90	19	9066	69	1
16	1360	90	20	12	36450	93	0
17	1075	49	67	25	8704	67	1
18	1240	95	40	17	15140	78	2
19	1290	75	50	13	38380	87	2
20	1180	65	68	16	15470	85	1
21	1285	80	36	11	27553	90	2
22	1225	77	44	14	13349	92	2
23	1085	40	69	15	11857	71	1
24	1375	95	19	11	43514	96	0

In [18]:
```
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(pdf, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()
```



In [19]:
```
from scipy.cluster.hierarchy import cophenet
import scipy.cluster.hierarchy as sch
from scipy.spatial.distance import pdist
```

In [21]:
```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=3,affinity='euclidean', linkage='com
test = cluster.fit_predict(pdf)
test
```

Out[21]:
```
array([0, 0, 2, 2, 2, 0, 0, 2, 0, 0, 0, 2, 2, 1, 0, 1, 0, 1, 2, 2, 1, 2,
       2, 1, 0], dtype=int64)
```

In [23]:

```
df['clusters']=pd.Series(test)
df
```

Out[23]:

	SAT	Top10	Accept	SFRatio	Expenses	GradRate	clusters
0	1310	89	22	13	22704	94	0
1	1415	100	25	6	63575	81	0
2	1260	62	59	9	25026	72	2
3	1310	76	24	12	31510	88	2
4	1280	83	33	13	21864	90	2
5	1340	89	23	10	32162	95	0
6	1315	90	30	12	31585	95	0
7	1255	74	24	12	20126	92	2
8	1400	91	14	11	39525	97	0
9	1305	75	44	7	58691	87	0
10	1380	94	30	10	34870	91	0
11	1260	85	39	11	28052	89	2
12	1255	81	42	13	15122	94	2
13	1081	38	54	18	10185	80	1
14	1375	91	14	8	30220	95	0
15	1005	28	90	19	9066	69	1
16	1360	90	20	12	36450	93	0
17	1075	49	67	25	8704	67	1
18	1240	95	40	17	15140	78	2
19	1290	75	50	13	38380	87	2
20	1180	65	68	16	15470	85	1
21	1285	80	36	11	27553	90	2
22	1225	77	44	14	13349	92	2
23	1085	40	69	15	11857	71	1
24	1375	95	19	11	43514	96	0

In [ ]: