Mall customer segmentaion: hierarchical clustering

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
data= pd.read_csv("mCustomers.csv")
data
```

Out[3]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [4]:

```
data.shape
```

Out[4]:

```
(200, 5)
```

In [5]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [10]:

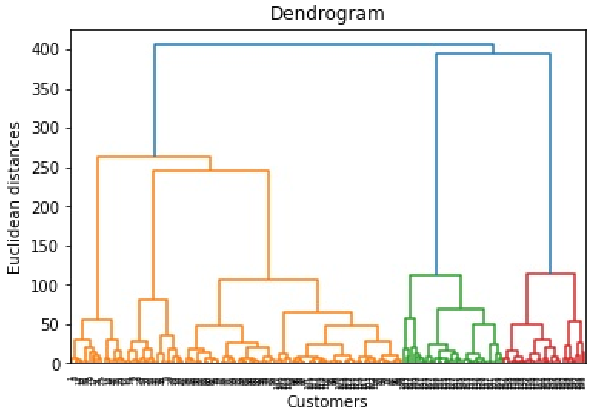
```
#retrieving all rows and columns 3(Annual income) and 4(Spending score) from the DataFrame df using integer-bas
x = data.iloc[:,[3,4]].values
```

In [14]:

```
import scipy.cluster.hierarchy as sch
```

In [26]:

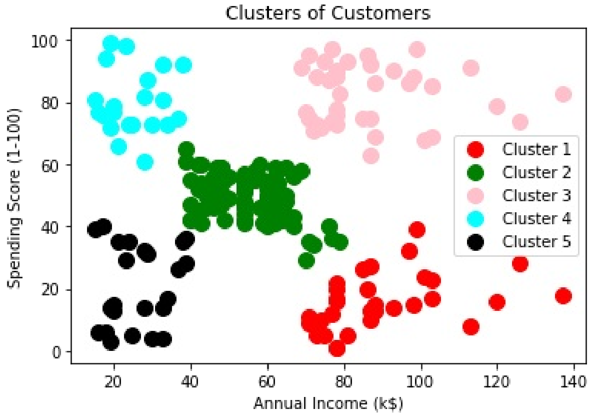
```
dendrogram = sch.dendrogram(sch.linkage(x, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()
```



In [20]:

```
from sklearn.cluster import  AgglomerativeClustering
hc = AgglomerativeClustering (n_clusters=5, affinity='euclidean', linkage='ward')
y_hc = hc.fit_predict(x)

plt.scatter(x[y_hc == 0,0], x[y_hc == 0,1], s=100, c='red', label='Cluster 1')
plt.scatter(x[y_hc == 1,0], x[y_hc == 1,1], s=100, c='green', label='Cluster 2')
plt.scatter(x[y_hc == 2,0], x[y_hc == 2,1], s=100, c='pink', label='Cluster 3')
plt.scatter(x[y_hc == 3,0], x[y_hc == 3,1], s=100, c='cyan', label='Cluster 4')
plt.scatter(x[y_hc == 4,0], x[y_hc == 4,1], s=100, c='black', label='Cluster 5')
plt.title('Clusters of Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js