```
#1) Make each data point a single  point cluster-->That forms n
#2) Take the two closest data points that makes one cluster-->
#3)Take the two closest clusters that makes one cluster--> That
#4) Repeat this step till thers is one cluster


# Distance between two clusters
#1)Closest point (Nearest points of two clusters)
#2)Furthest point (farest point of clusters)
#3)Average distance (average distance of clusters)
#4)Distance between the centroids


#import the libraries (for more information press shift+enter)
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt



df=pd.read_csv('Mall_Customers.csv')
df.values
```

```
array([[1, 'Male', 19, 15, 39],
       [2, 'Male', 21, 15, 81],
       [3, 'Female', 20, 16, 6],
       [4, 'Female', 23, 16, 77],
       [5, 'Female', 31, 17, 40],
       [6, 'Female', 22, 17, 76],
       [7, 'Female', 35, 18, 6],
       [8, 'Female', 23, 18, 94],
       [9, 'Male', 64, 19, 3],
       [10, 'Female', 30, 19, 72],
       [11, 'Male', 67, 19, 14],
       [12, 'Female', 35, 19, 99],
       [13, 'Female', 58, 20, 15],
       [14, 'Female', 24, 20, 77],
       [15, 'Male', 37, 20, 13],
       [16, 'Male', 22, 20, 79],
       [17, 'Female', 35, 21, 35],
       [18, 'Male', 20, 21, 66],
       [19, 'Male', 52, 23, 29],
       [20, 'Female', 35, 23, 98],
       [21, 'Male', 35, 24, 35],
       [22, 'Male', 25, 24, 73],
       [23, 'Female', 46, 25, 5],
       [24, 'Male', 31, 25, 73],
       [25, 'Female', 54, 28, 14],
       [26, 'Male', 29, 28, 82],
       [27, 'Female', 45, 28, 32],
       [28, 'Male', 35, 28, 61],
       [29, 'Female', 40, 29, 31],
```

```
[30, 'Female', 23, 29, 87],
[31, 'Male', 60, 30, 4],
[32, 'Female', 21, 30, 73],
[33, 'Male', 53, 33, 4],
[34, 'Male', 18, 33, 92],
[35, 'Female', 49, 33, 14],
[36, 'Female', 21, 33, 81],
[37, 'Female', 42, 34, 17],
[38, 'Female', 30, 34, 73],
[39, 'Female', 36, 37, 26],
[40, 'Female', 20, 37, 75],
[41, 'Female', 65, 38, 35],
[42, 'Male', 24, 38, 92],
[43, 'Male', 48, 39, 36],
[44, 'Female', 31, 39, 61],
[45, 'Female', 49, 39, 28],
[46, 'Female', 24, 39, 65],
[47, 'Female', 50, 40, 55],
[48, 'Female', 27, 40, 47],
[49, 'Female', 29, 40, 42],
[50, 'Female', 31, 40, 42],
[51, 'Female', 49, 42, 52],
[52, 'Male', 33, 42, 60],
[53, 'Female', 31, 43, 54],
[54, 'Male', 59, 43, 60],
[55, 'Female', 50, 43, 45],
[56, 'Male', 47, 43, 41],
[57, 'Female', 51, 44, 50],
[58, 'Male', 69, 44, 46],
```

df

	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

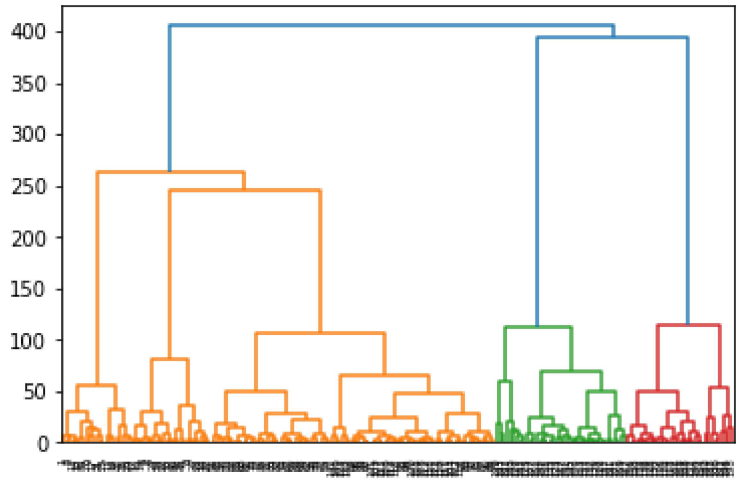
200 rows × 5 columns

```
x=df.iloc[:,3:].values
```

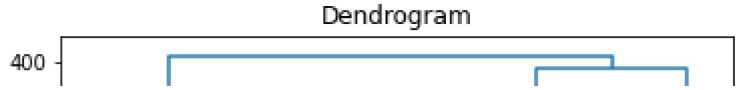
X

```
array([[ 15,  39],
       [ 15,  81],
       [ 16,   6],
       [ 16,  77],
       [ 17,  40],
       [ 17,  76],
       [ 18,   6],
       [ 18,  94],
       [ 19,   3],
       [ 19,  72],
       [ 19,  14],
       [ 19,  99],
       [ 20,  15],
       [ 20,  77],
       [ 20,  13],
       [ 20,  79],
       [ 21,  35],
       [ 21,  66],
       [ 23,  29],
       [ 23,  98],
       [ 24,  35],
       [ 24,  73],
       [ 25,   5],
       [ 25,  73],
       [ 28,  14],
       [ 28,  82],
       [ 28,  32],
       [ 28,  61],
       [ 29,  31],
       [ 29,  87],
       [ 30,   4],
       [ 30,  73],
       [ 33,   4],
       [ 33,  92],
       [ 33,  14],
       [ 33,  81],
       [ 34,  17],
       [ 34,  73],
       [ 37,  26],
       [ 37,  75],
       [ 38,  35],
       [ 38,  92],
       [ 39,  36],
       [ 39,  61],
       [ 39,  28],
       [ 39,  65],
       [ 40,  55],
       [ 40,  47],
       [ 40,  42],
       [ 40,  42],
       [ 42,  52],
       [ 42,  60],
       [ 43,  54],
       [ 43,  60],
       [ 43,  45],
       [ 43,  41],
       [ 44,  50],
```

```
#Dendogram to find the optimal no of clusters
import scipy.cluster.hierarchy as sch
dendrogram=sch.dendrogram(sch.linkage(x,'ward')) #method='ward'
```



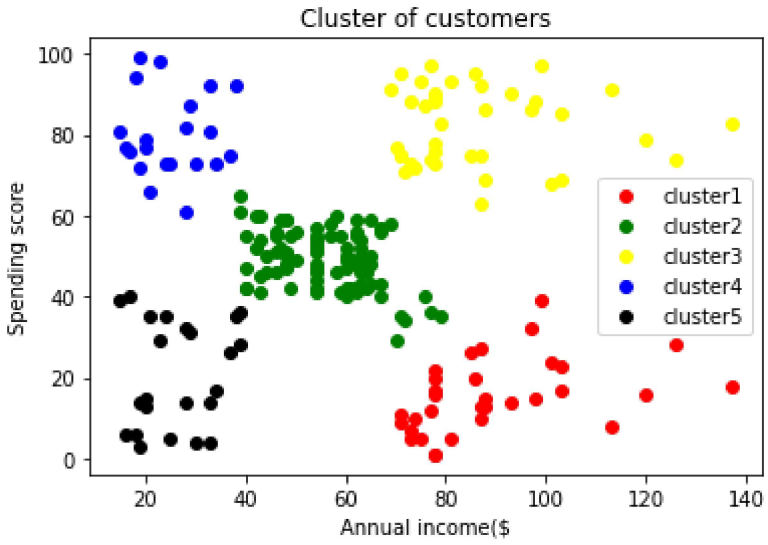

```
#This above diagram called as dendrogram
#The distance between each datapoint is Euclidean distance

#Google the scipy.cluster.hierarchy function
#scipy sch documentation

#Google the AgglomerativeClustering

import scipy.cluster.hierarchy as sch
dendrogram=sch.dendrogram(sch.linkage(x,'ward')) #method='ward'
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distance')
plt.show()
```




```
#Train the model
from sklearn.cluster import AgglomerativeClustering
clustering=AgglomerativeClustering(n_clusters=5)
y=clustering.fit_predict(x)
y
```

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

```
#Visualizing the data
plt.scatter(x[y==0,0],x[y==0,1],color='red',label='cluster1')
plt.scatter(x[y==1,0],x[y==1,1],color='green',label='cluster2')
plt.scatter(x[y==2,0],x[y==2,1],color='yellow',label='cluster3'
plt.scatter(x[y==3,0],x[y==3,1],color='blue',label='cluster4')
plt.scatter(x[y==4,0],x[y==4,1],color='black',label='cluster5')
plt.title('Cluster of customers')
plt.xlabel('Annual income($')
plt.ylabel('Spending score')
plt.legend()
plt.show()
```



```
#RED Color represents high income and high spending score
#BLACK Color represents less income and also less spending scor
```

```
from sklearn.cluster import AgglomerativeClustering
clustering=AgglomerativeClustering(n_clusters=6)
y=clustering.fit_predict(x)
y
```

```
array([4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3,
       4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 1,
       4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 5, 0, 5, 0,
       1, 0, 5, 0, 5, 0, 5, 0, 5, 0, 1, 0, 5, 0, 1, 0, 5, 0, 5, 0, 5, 0,
       5, 0, 5, 0, 5, 0, 1, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0,
       5, 0, 5, 0, 2, 0, 5, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 0,
       2, 0])
```

```
#Visualizing the data
plt.scatter(x[y==0,0],x[y==0,1],color='red',label='cluster1')
plt.scatter(x[y==1,0],x[y==1,1],color='green',label='cluster2')
plt.scatter(x[y==2,0],x[y==2,1],color='yellow',label='cluster3'
plt.scatter(x[y==3,0],x[y==3,1],color='blue',label='cluster4')
plt.scatter(x[y==4,0],x[y==4,1],color='black',label='cluster5')
plt.scatter(x[y==5,0],x[y==5,1],color='orange',label='cluster6'
plt.title('Cluster of customers')
plt.xlabel('Annual income($')
plt.ylabel('Spending score')
plt.legend()
plt.show()
```

'''k-means is method of cluster analysis using a pre-specified
It requires advance knowledge of ‘K’.'''

'''Hierarchical clustering also known as hierarchical cluster a

'Hierarchical clustering also known as hierarchical cluster analysis (HCA) is also a
of cluster analysis which seeks to build a hierarchy of clusters without having fixe