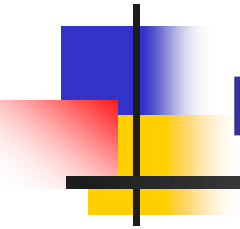
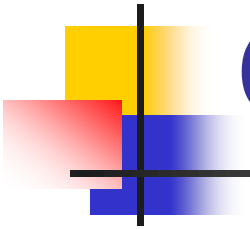
Introduction to Data Science CS61 June 12 - July 12, 2018



Dr. Ash Pahwa

Lesson 10: Clustering

Lesson 10.1: Hierarchical Clustering



Outline

- What is Hierarchical Clustering
- Strategy for Hierarchical Clustering
- Dendrograms
- Algorithm for Hierarchical Clustering
- Distance between Clusters
- Hierarchical Clustering in R
- Hierarchical Clustering in Python

What is Hierarchical Clustering?

- Disadvantage of K-means clustering
  - You have specify the number of clusters
- Hierarchical clustering solves this problem – no specification of number of clusters
- Hierarchical structure also creates a hierarchical structure of data called
  - Dendrogram

Strategy to build Hierarchical Clustering

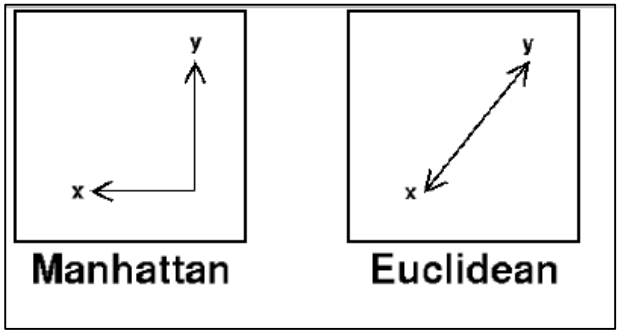
- Bottom-up approach
  - Agglomerative clustering
- Compute the Euclidean distance between data points
  - Shortest distance observations should be in a single cluster
  - Next we compute the distance between cluster that we have created and the next point closets to it
  - Include that point in that cluster

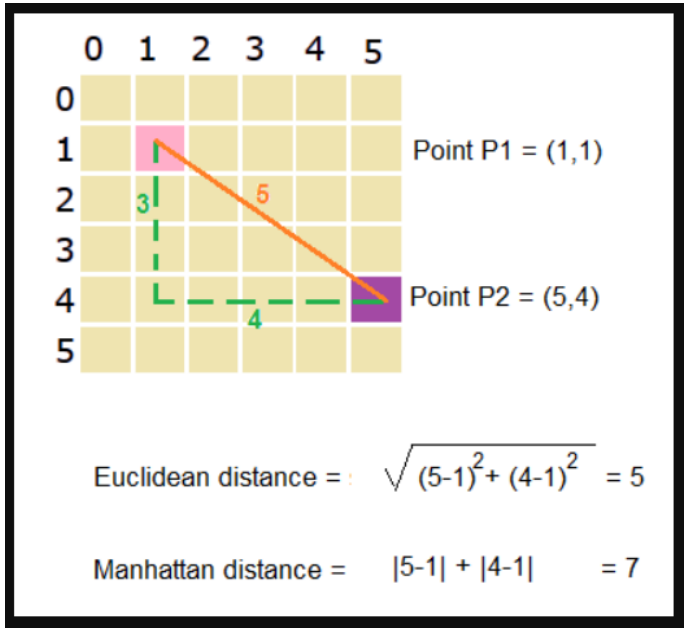
Distance Between Observations

- Compute the distance matrices between objects

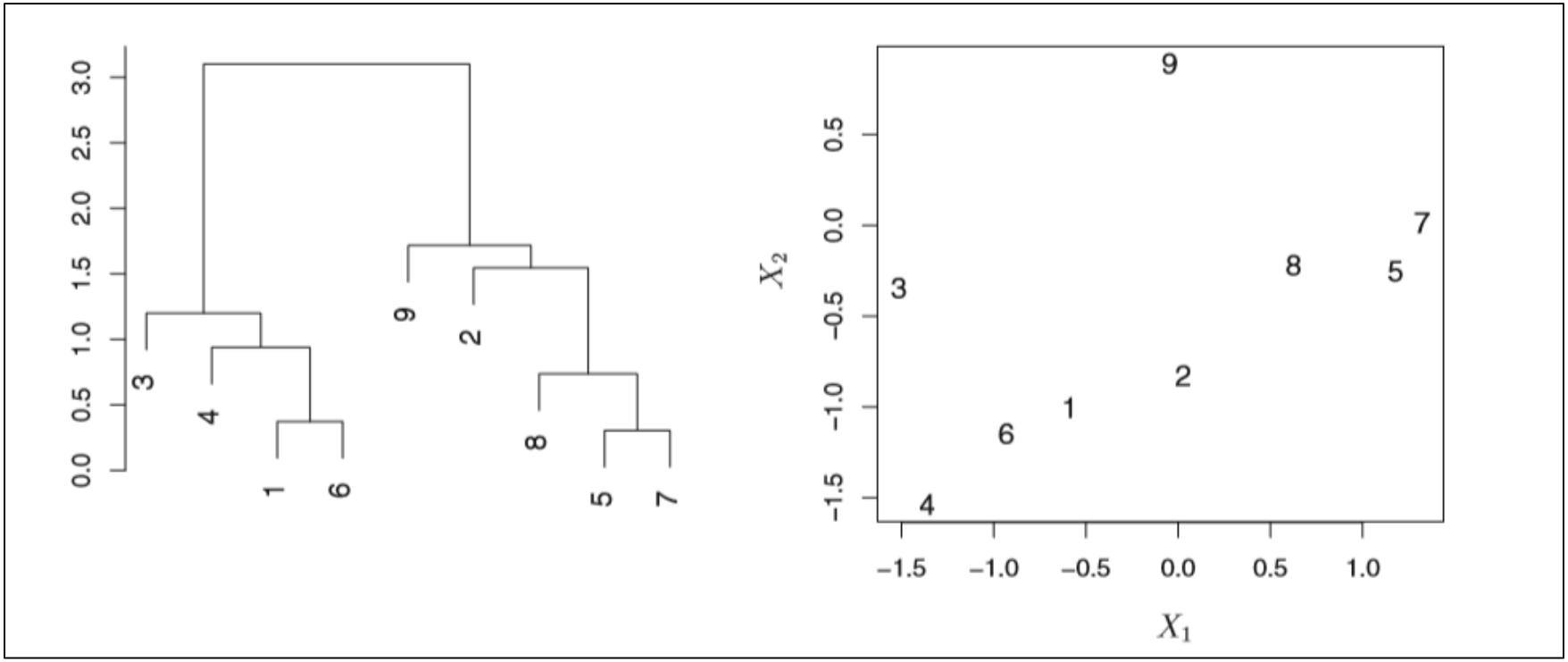
$$Euclidean\ distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$Manhattan\ distance = |x_2 - x_1| + |y_2 - y_1|$$

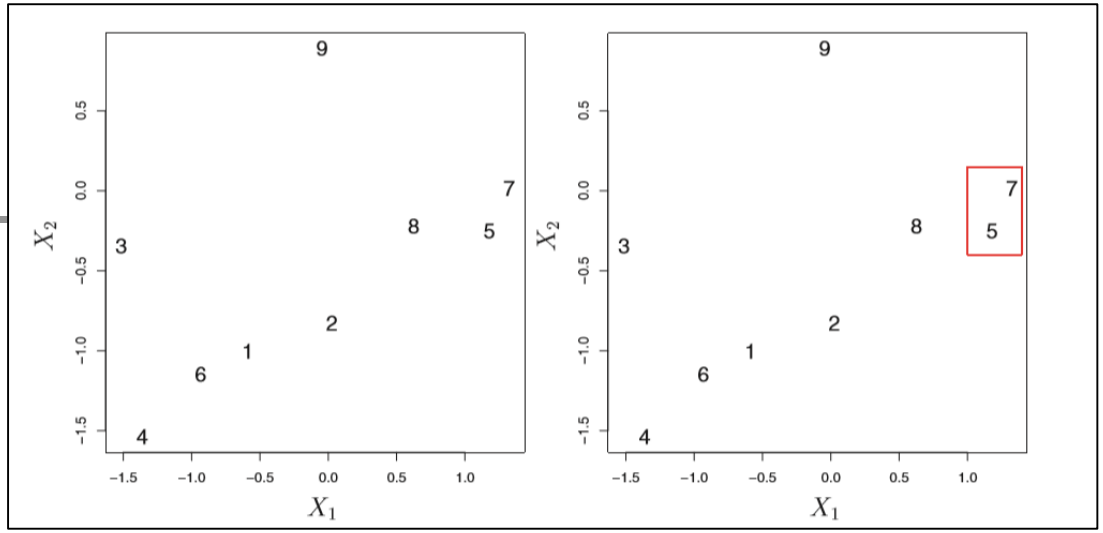


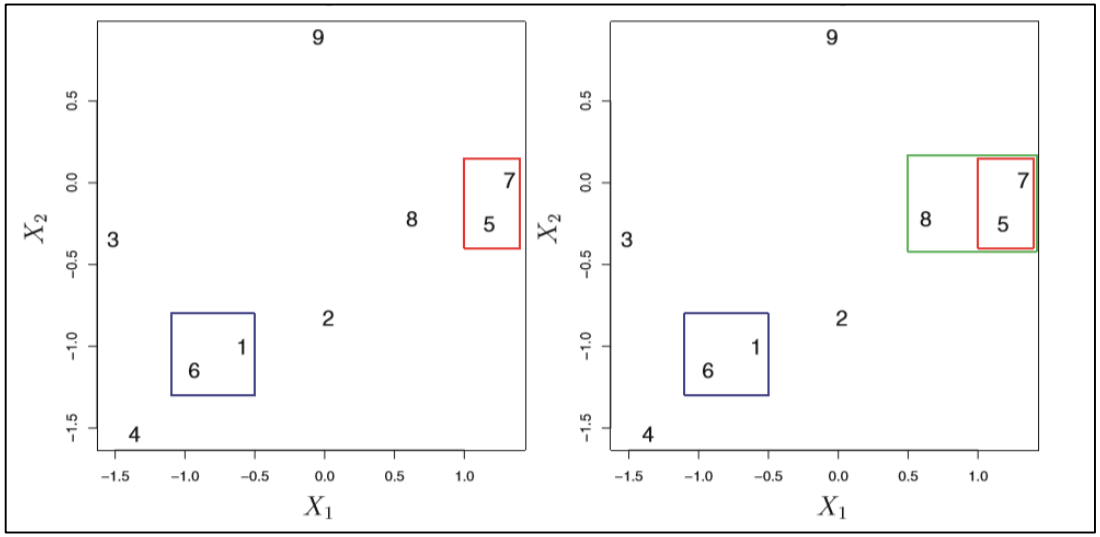


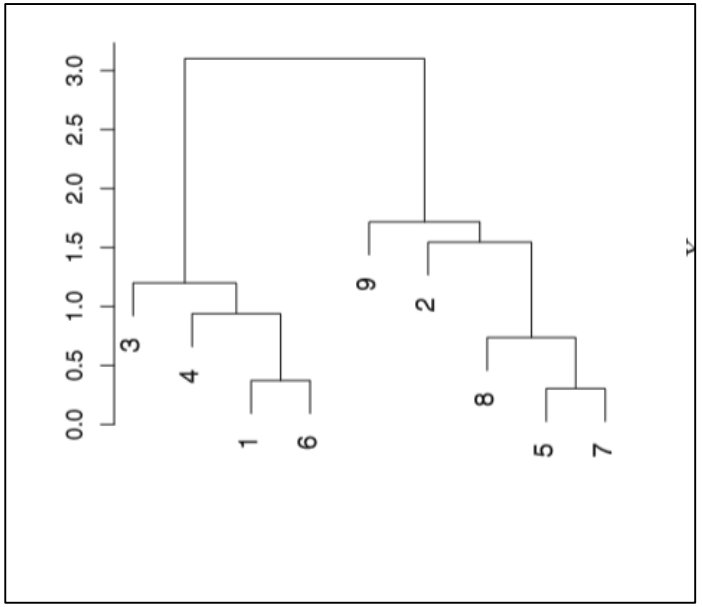
Hierarchical Clustering Example



Step-By-Step

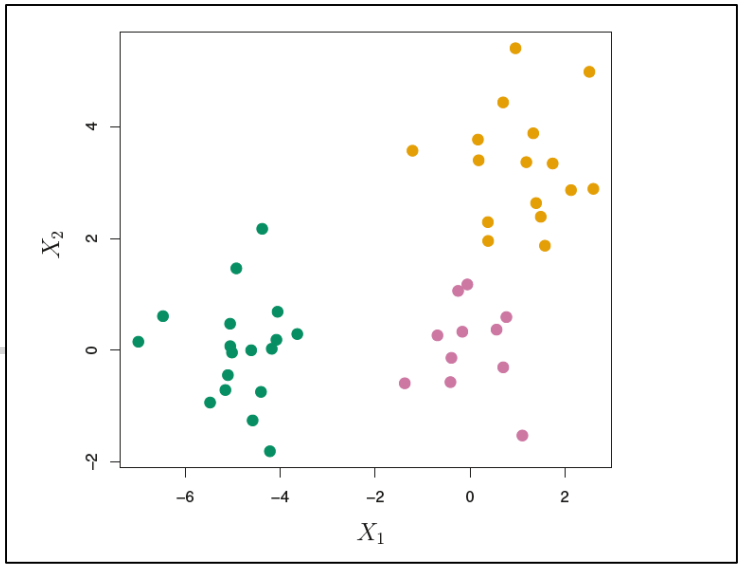


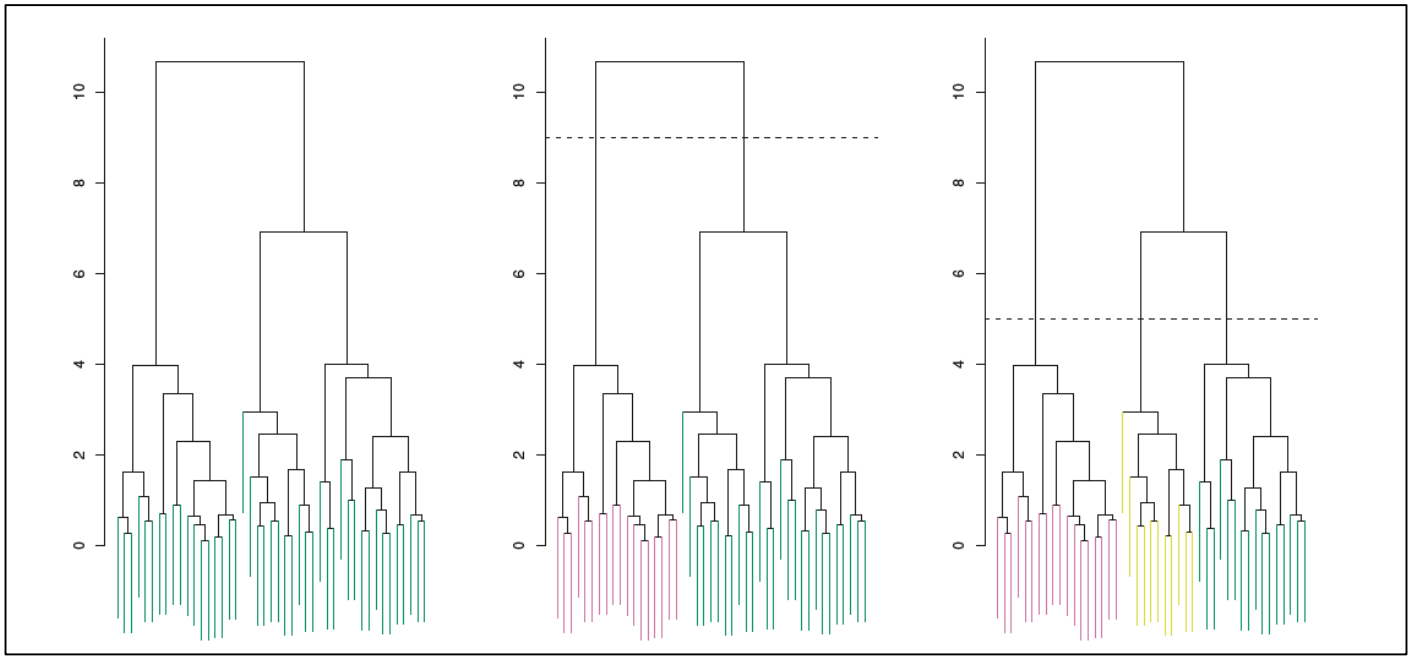




Dendrogram

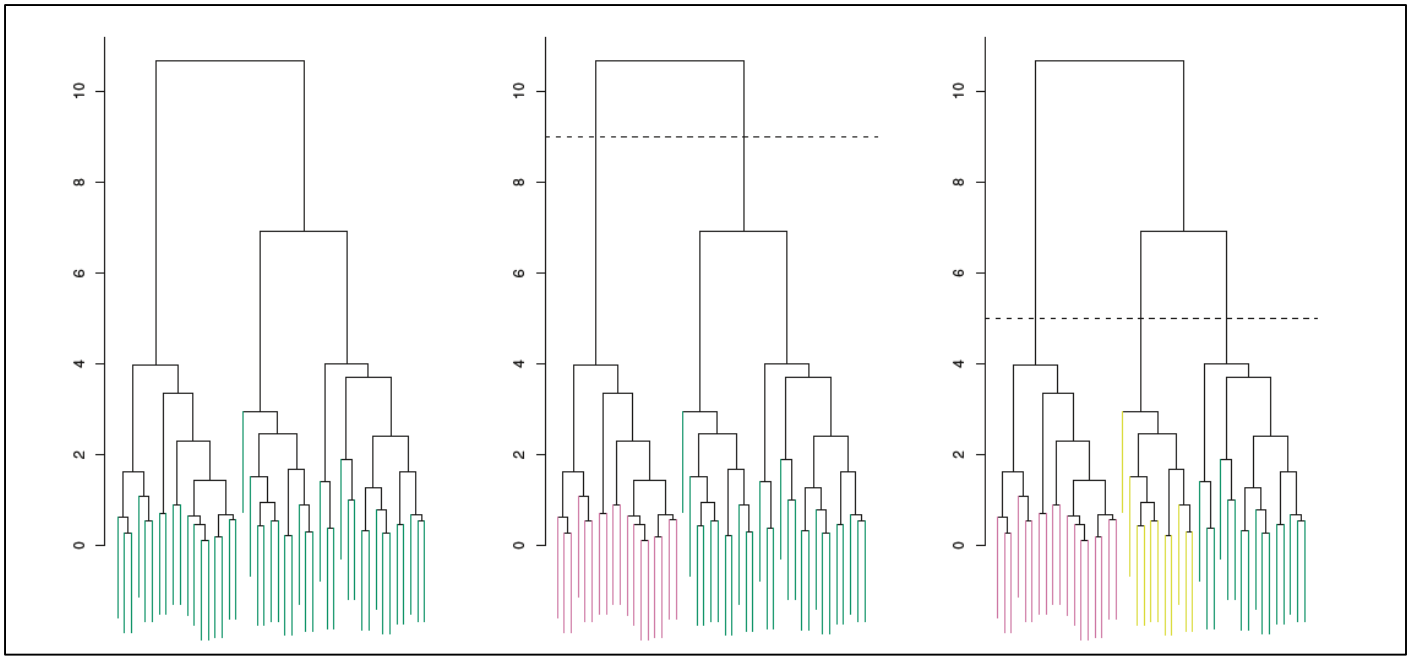
- To Create a cluster
  - Draw a horizontal cut across the dendrogram





Dendrogram

- The height of a dendrogram is equivalent to the 'k' of the k-means clustering



1 Cluster

2 Clusters

3 Clusters

Hierarchical Clustering Algorithm

Algorithm 10.2 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

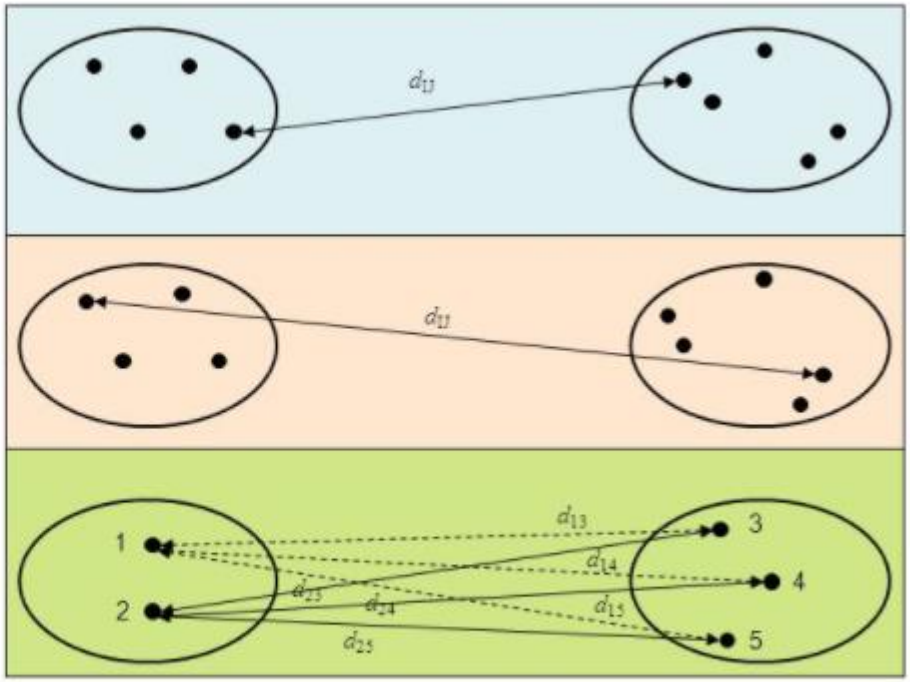
   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

How to Compute the Distance between Clusters

Methods to define a distance between clusters:

single linkage:

$$d_{IJ} = \min\{ d_{ij} : i \in I \text{ and } j \in J \}$$



complete linkage:

$$d_{IJ} = \max\{ d_{ij} : i \in I \text{ and } j \in J \}$$

group average:

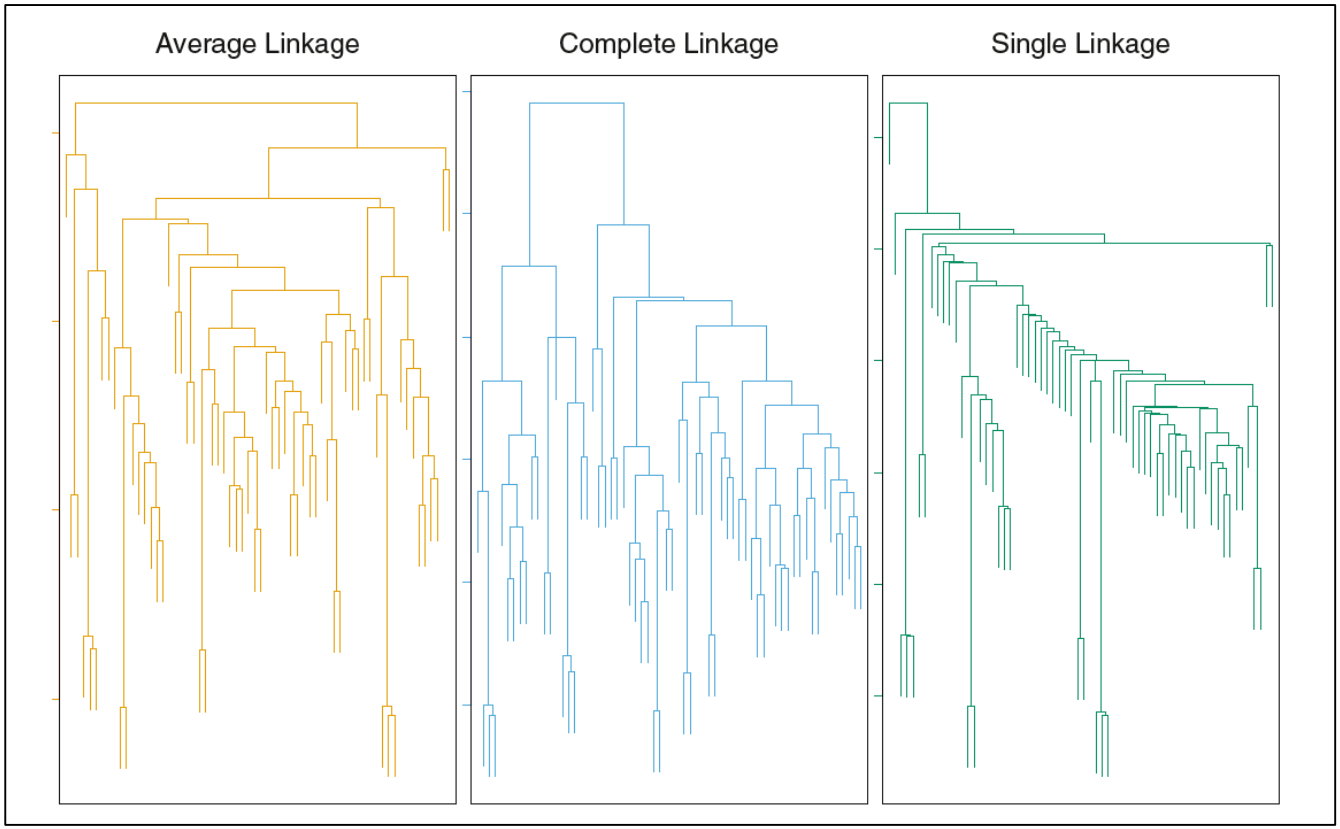
$$d_{IJ} = \sum_{i \in I} \sum_{j \in J} d_{ij} / (N_I N_J)$$

N is the number of members in a cluster

centroid linkage:

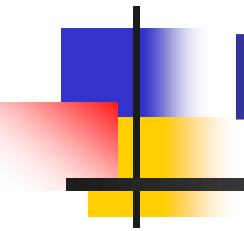
$$d_{IJ} = d(\bar{x}_I, \bar{x}_J) \text{ where } \bar{x}_J = \frac{1}{n_J} \sum_{i=1}^{n_J} x_{iJ}$$

Average Linkage tends to Create Balanced Dendrogram



Distance between Clusters Correlation

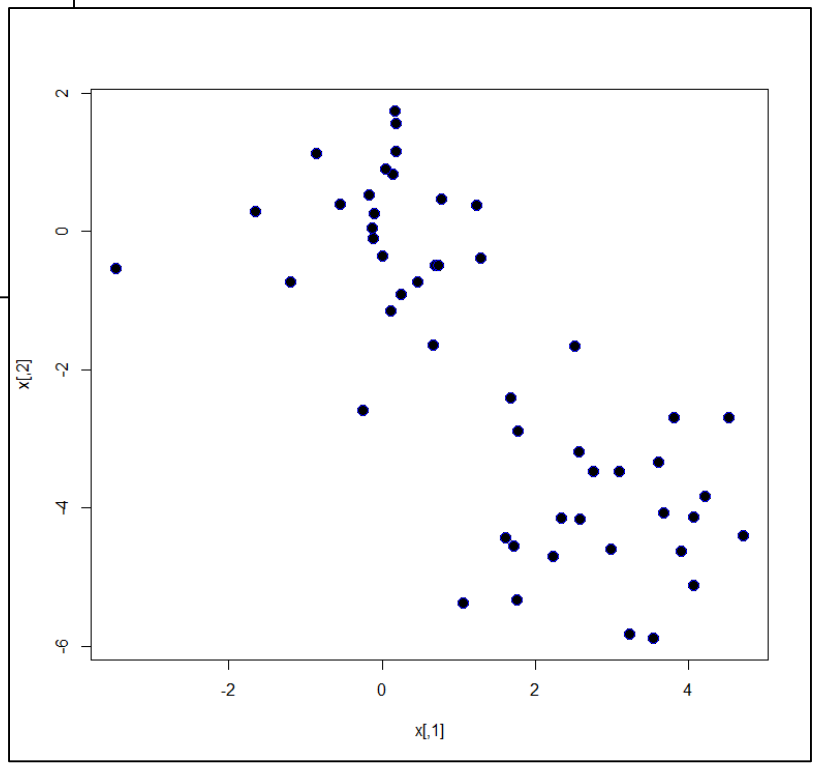
- Distance between Clusters can also be Computed by Correlation between 2 sets of objects in each cluster



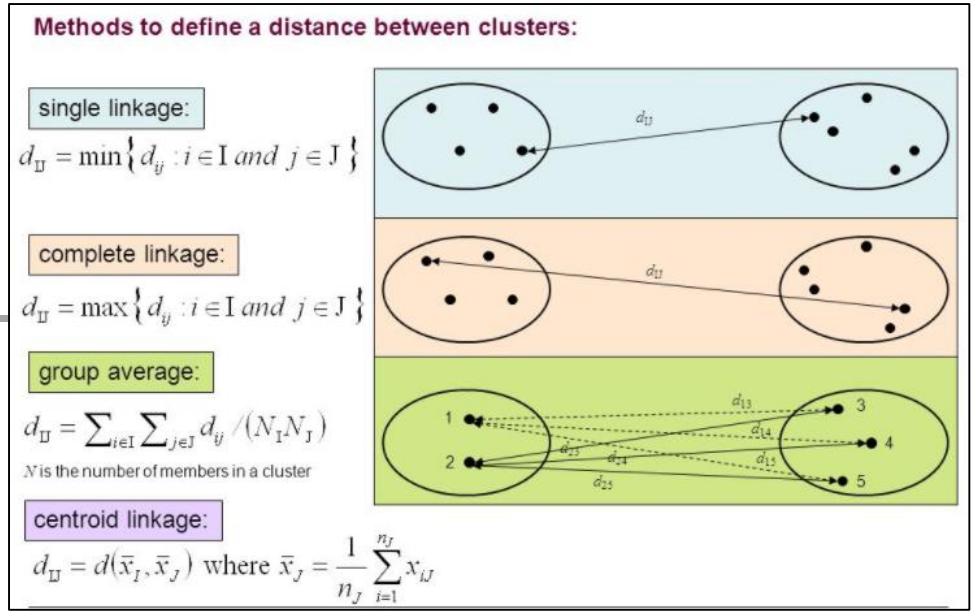
Hierarchical Clustering in R

Dataset

```
> ################################################
> # 0. Generate Dataset
> #
> x=matrix(rnorm(50*2), ncol=2)
> #x
> x[1:25,1]=x[1:25,1]+3
> x[1:25,2]=x[1:25,2]-4
> plot(x,pch=21,col='blue',bg='black',cex=1.5)
>
```

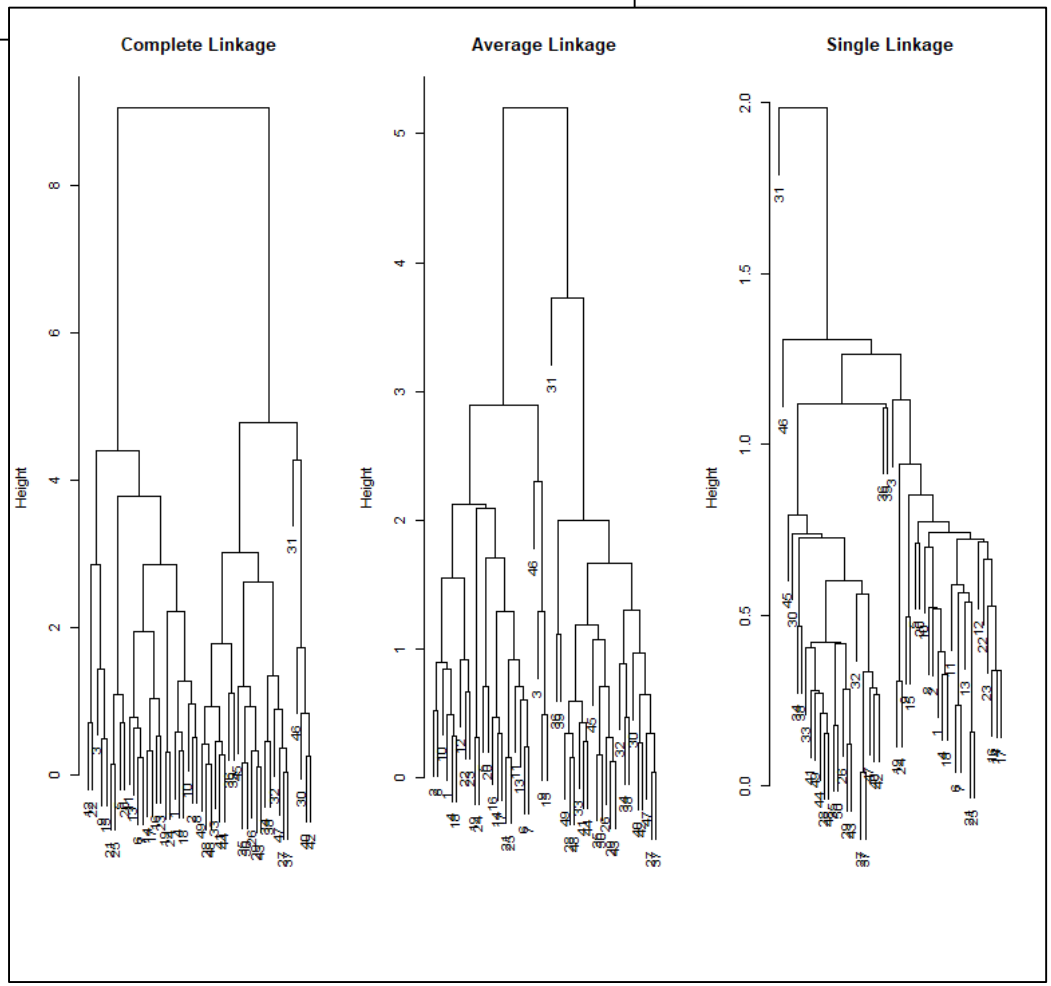


Create 3 Trees




```
> ###################################################
> # 1 + 2: Create 3 Trees
> #
> hc.complete=hclust(dist(x), method="complete")
> hc.average=hclust(dist(x), method="average")
> hc.single=hclust(dist(x), method="single")
>
>
```

```
> ###################################################
> # 3. Plot 3 Trees
> #
> par(mfrow=c(1,3))
> plot(hc.complete,main="Complete Linkage", xlab="", sub="", cex=.9)
> plot(hc.average, main="Average Linkage", xlab="", sub="", cex=.9)
> plot(hc.single, main="Single Linkage", xlab="", sub="", cex=.9)
>
```

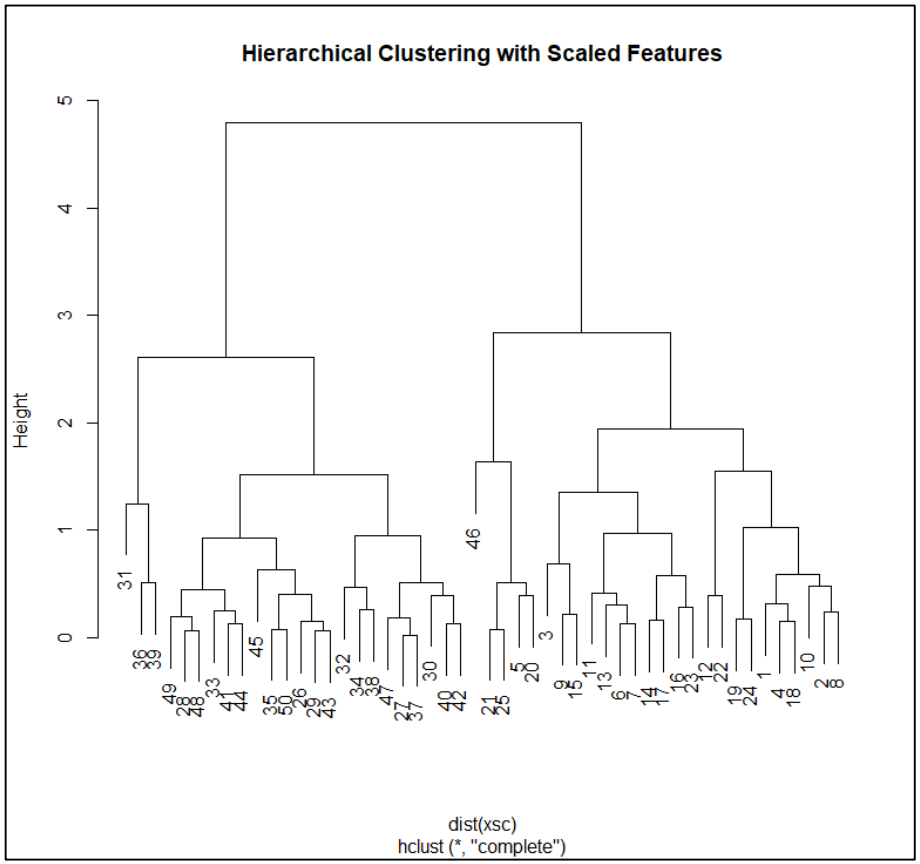


Cut Trees

```
> ############################################
> # 4. Cut Trees
> #
> cutree(hc.complete, 2)
 [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 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2
> cutree(hc.average, 2)
 [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 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 1 2 2 2 2
> cutree(hc.single, 2)
 [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 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
>
> ############################################
> # 5. Cut Tree
> #
> cutree(hc.single, 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 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2
2 2 2 4 2 2 2 2
>
```

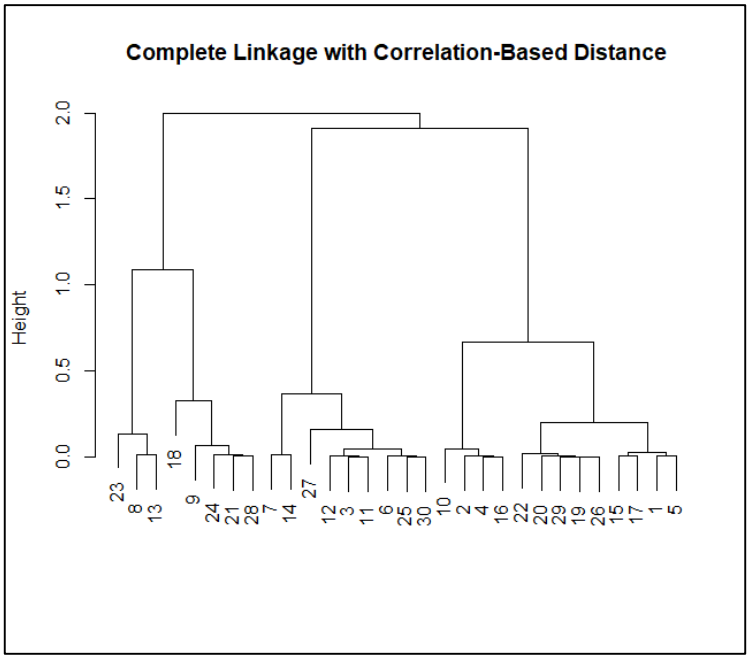
Scale the values and then Create Clusters

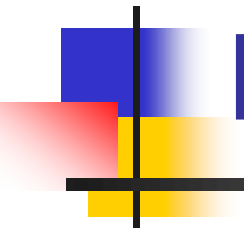
```
> ##########################################
> # 6. Scaling
> #
> xsc=scale(x)
> par(mfrow=c(1,1))
> plot(hclust(dist(xsc), method="complete"),
main="Hierarchical Clustering with Scaled
Features")
>
```



Distance between Clusters = Correlation

```
> ################################################
> # 7. Correlation
> #
> x=matrix(rnorm(30*3), ncol=3)
> dd=as.dist(1-cor(t(x)))
> plot(hclust(dd, method="complete"), main="Complete Linkage with Correlation-
Based Distance", xlab="", sub="")
>
```





Hierarchical Clustering in Python

Dataset

	A	B	C	D	E	F
1	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	
2	1	Male	19	15	39	
3	2	Male	21	15	81	
4	3	Female	20	16	6	
5	4	Female	23	16	77	
6	5	Female	31	17	40	
7	6	Female	22	17	76	
8	7	Female	35	18	6	
9	8	Female	23	18	94	
10	9	Male	64	19	3	
11	10	Female	30	19	72	
12	11	Male	67	19	14	
13	12	Female	35	19	99	
14	13	Female	58	20	15	
15	14	Female	24	20	77	
16	15	Male	37	20	13	
17	16	Male	22	20	79	
18	17	Female	35	21	35	
19	18	Male	20	21	66	
20	19	Male	52	23	29	
21	20	Female	35	23	98	
22	21	Male	35	24	35	

Load the Libraries and Read the Dataset

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

##########################################################

dataset = pd.read_csv("Mall_Customers.csv")

len(dataset)
Out[5]: 200
```

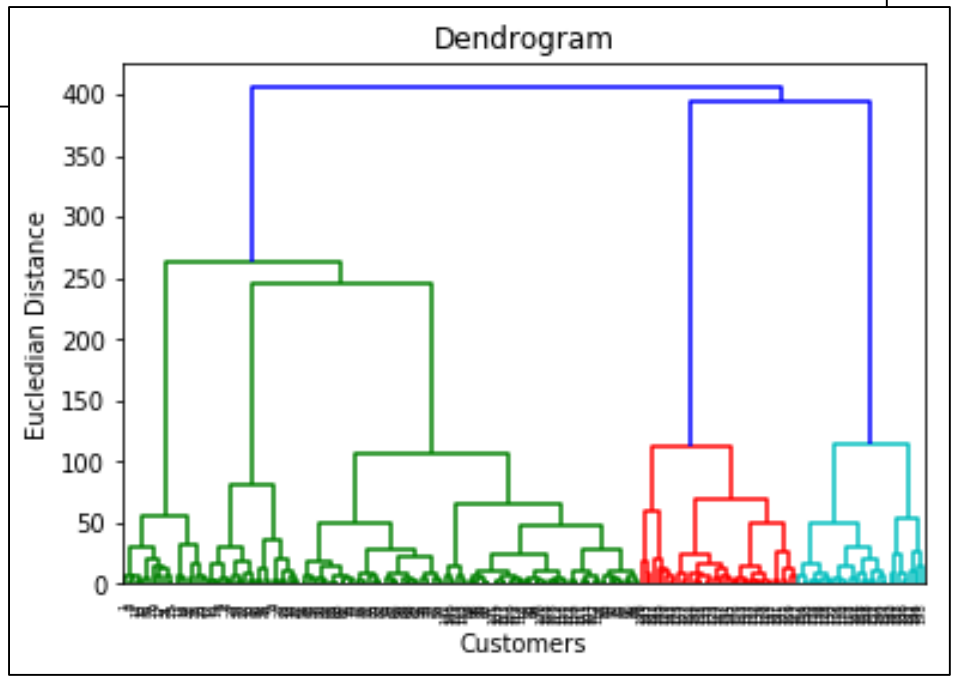
Retrieve Annual Income (3rd Column) and Spending Score (4th Column) from Dataset

```
############################################################
# Since we are performing clustering
# we only need X variable
#
# Clustering is a unsupervised method,
# that's why we do NOT need the response variable or the 'y' variable
#

X = dataset.iloc[:,[3,4]].values
```

	A	B	C	D	E	F
1	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	
2	1	Male	19	15	39	
3	2	Male	21	15	81	
4	3	Female	20	16	6	

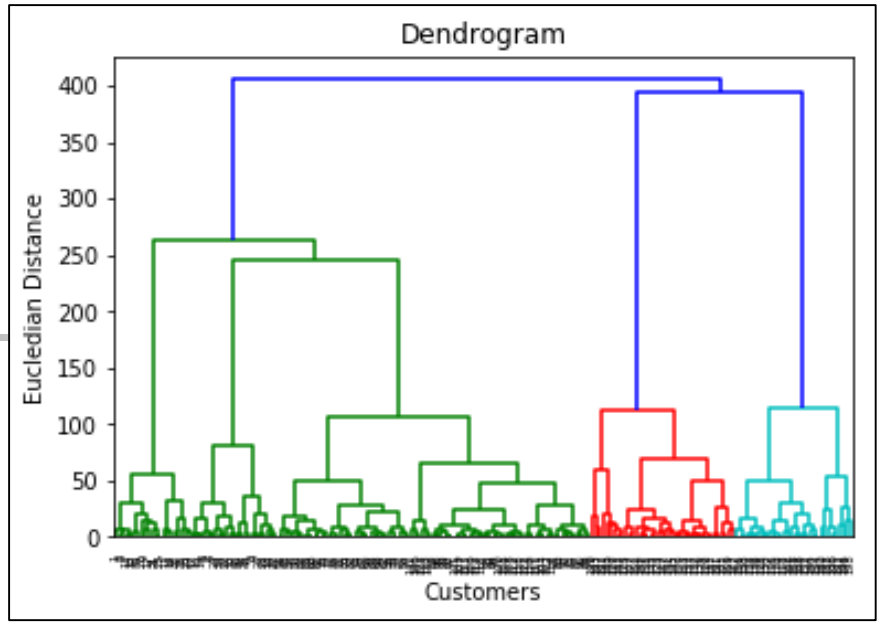
Plot the Dendrogram

```
#############################################################
# Plot the dendrogram
# The plot will determine how many clusters we should need
#

import scipy.cluster.hierarchy as sch

dendrogram = sch.dendrogram(sch.linkage(X,method='ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Eucledian Distance')
```



Ward's minimum variance method is a special case of the objective function approach originally presented by Joe H. Ward, Jr.

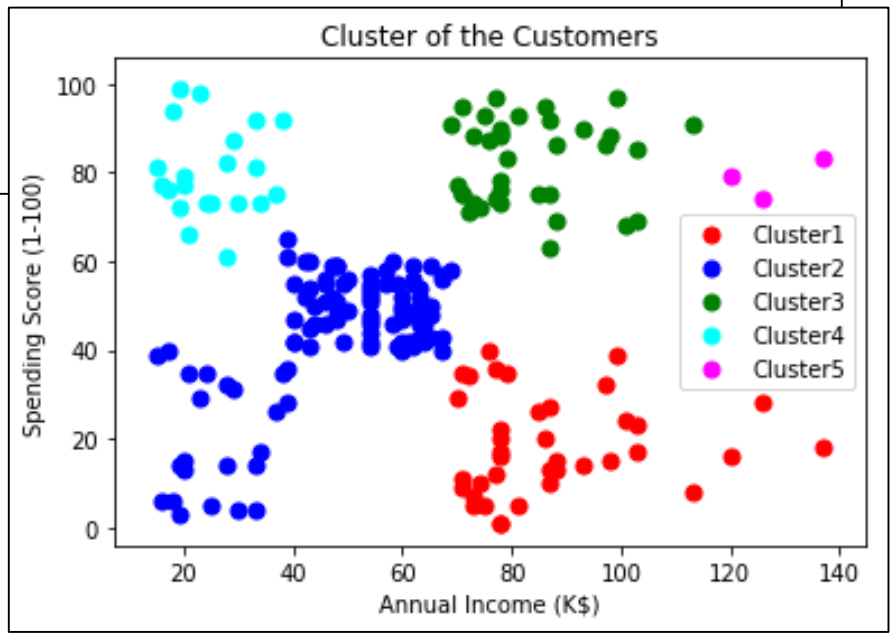
Create Clusters
Predict the Clusters




```
# We can have 3 clusters as standard
# Or we can have 5 clusters
# Find the longest line which is not crossed by horizontal line
#
# This shows total number of clusters = 5
#

from sklearn.cluster import AgglomerativeClustering

hc = AgglomerativeClustering(n_clusters=5, affinity='euclidean',linkage='average')

y_hc = hc.fit_predict(X)
```

Plot the Clusters

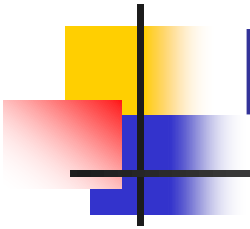
```
plt.scatter(X[y_hc==0,0],X[y_hc==0,1],s=50,c='red',label='Cluster1')
plt.scatter(X[y_hc==1,0],X[y_hc==1,1],s=50,c='blue',label='Cluster2')
plt.scatter(X[y_hc==2,0],X[y_hc==2,1],s=50,c='green',label='Cluster3')
plt.scatter(X[y_hc==3,0],X[y_hc==3,1],s=50,c='cyan',label='Cluster4')
plt.scatter(X[y_hc==4,0],X[y_hc==4,1],s=50,c='magenta',label='Cluster5')

plt.title('Cluster of the Customers')
plt.xlabel('Annual Income (K$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
```



- Five Clusters of People
  - Low Income, Low Spending
  - Medium Income, Medium Spending
  - High Income, High Spending
  - Low Income, High Spending
  - High Income, Low Spending

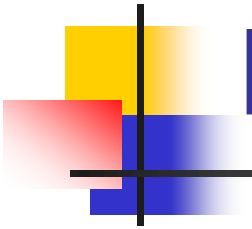
Practical Issues in Clustering

- Decisions have to be taken
  - K-means: what should be the value of 'k'
  - Hierarchical clustering
    - How to compute distance between 2 data points
    - How to compute distance between clusters
    - Where should we cut the dendrogram tree
- Based on these decisions different clusters will be obtained



Practical Issues in Clustering

- Any random set of data will generate clusters
- How do we know whether we have identified a cluster?
  - Measure ‘Within-cluster-variation’

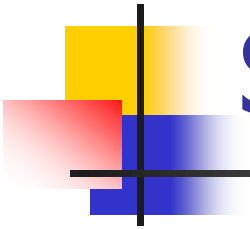


Practical Issues in Clustering

- Outliers distort the boundaries of clusters
- Any small change in the data should not change the clusters
  - But sometimes it does

Practical Issues in Clustering

- Clustering results are not precise
  - Results will vary with change in the clustering parameters
  - Run clustering algorithm with many parameter values and analyze full set of results
- Clustering decreases the size and complexity of problems for other data mining methods



Summary

- What is Hierarchical Clustering
- Strategy for Hierarchical Clustering
- Dendrograms
- Algorithm for Hierarchical Clustering
- Distance between Clusters
- Hierarchical Clustering in R
- Hierarchical Clustering in Python