Chapter 6: Clustering

- K - Means Clustering
- Hierarchical Clustering

Clustering - K - Means Clustering

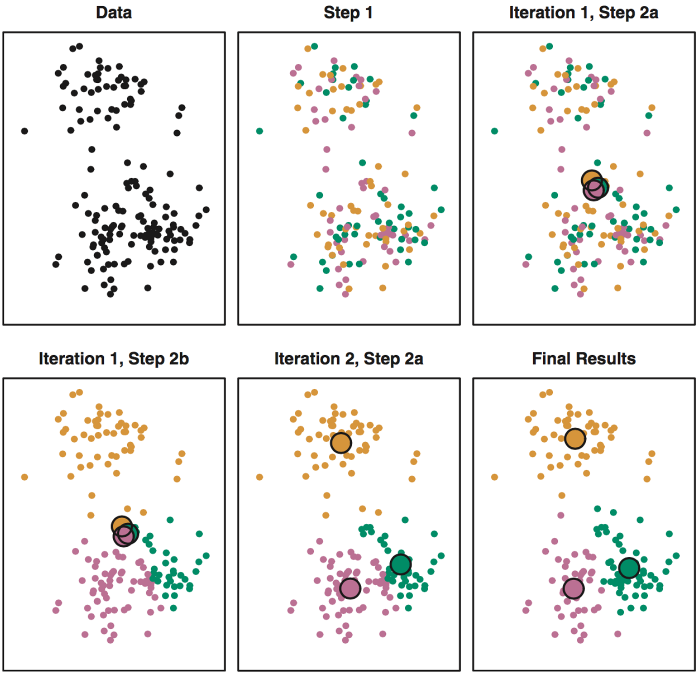
The idea behind K - means clustering is that a good clustering is one for which the within-cluster variation is as small as possible. We want to minimise

$$minimise_{C_1,\ldots,C_K}\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{i,j}-x_{i'j})^2\right\},$$

where C_k denotes the number of observations in the kth cluster.

K - Means Clustering Algorithm:

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).

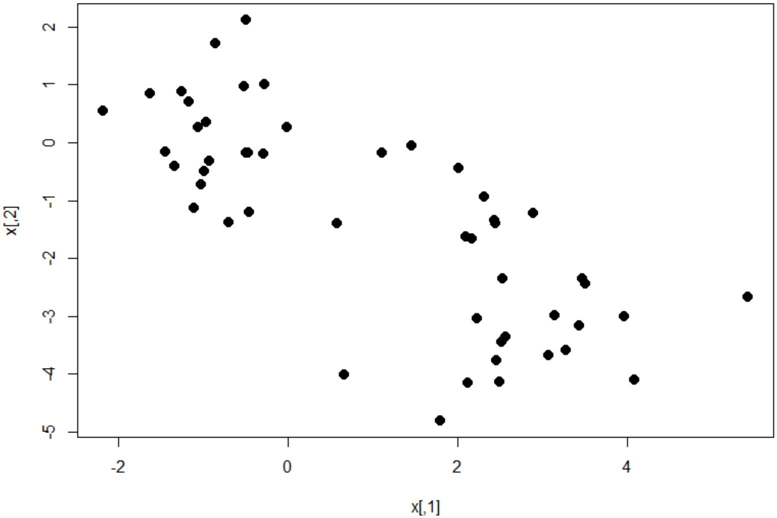

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Clustering - K - Means Clustering

The function kmeans() performs K-means clustering in R. We begin with a simple simulated example in which there truly are two clusters in the data: the first 25 observations have a mean shift relative to the next 25 observations.

```
> set.seed(1234)
> x = matrix(data = rnorm(n = 100), ncol = 2)
>
> # for the first 25 obs, x1(x axis) + 3
> # for the first 25 obs, x2(y axis) − 3
> x[1:25,1] = x[1:25,1] + 3
> x[1:25,2] = x[1:25,2] − 3
> plot(x, pch = 20, cex = 2)
```

Clustering - K - Means Clustering



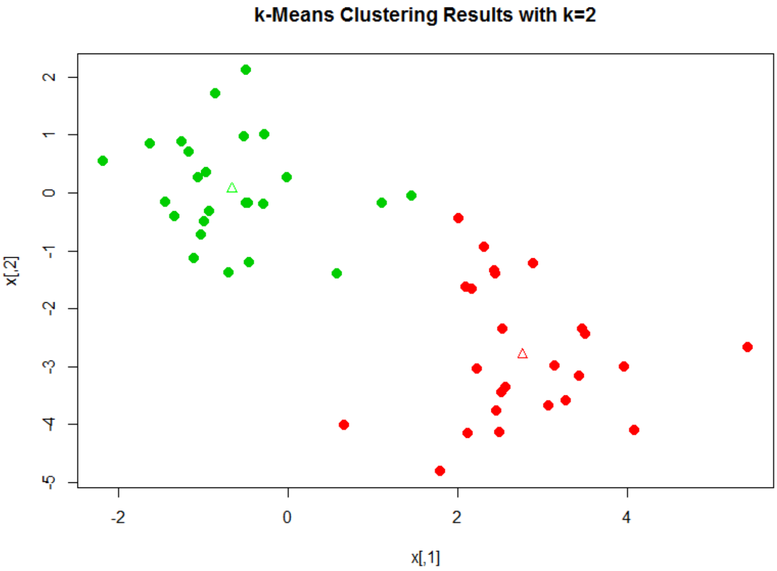
Clustering - K - Means Clustering

We strongly recommend always running k-means clustering with a large value of "nstart", such as 20 or 50, since otherwise an undesirable local optimum may be obtained.

```
> km_cluster = kmeans(x = x, centers = 2, nstart = 50)
> plot(x, col = (km_cluster$cluster + 1),
+       main = "k-Means Clustering Results with k=2",
+       pch = 20, cex = 2)
> points(km_cluster$centers, pch = 2, col = c("red", "green"))
```

The K-means clustering perfectly separated the observations into two clusters even though we did not supply any group information to kmeans(). We can plot the data, with each observation colored according to its cluster assignment.

Clustering - K - Means Clustering

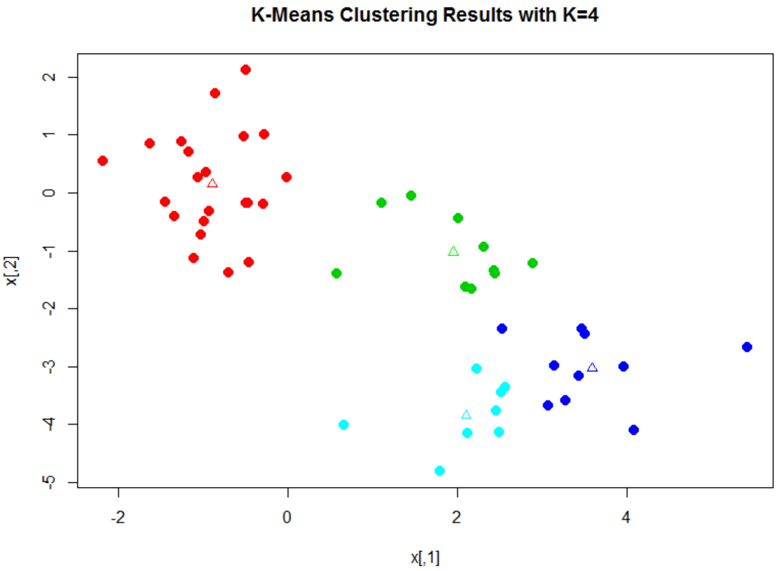


Clustering - K - Means Clustering

In last example, we knew that there really were two clusters because we generated the data. However, for real data, in general we do not know the true number of clusters. We could instead have performed K-means clustering on this example with K = 4.

```r
km_cluster = kmeans(x = x, centers = 4, nstart = 50)
plot(x, col = (km_cluster$cluster + 1),
     main = "K-Means Clustering Results with K=4",
     pch = 20, cex = 2)
points(km_cluster$centers, pch = 2,
       col = c("red", "green", "blue", "cyan"))
```

Clustering - K - Means Clustering



Clustering - Hierarchical Clustering

One potential disadvantage of K-means clustering is that it requires us to pre-specify the number of clusters K. Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K.

Hierarchical Clustering Algorithm:

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{2}{n} = n(n-1)/2$ 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.

Clustering - Hierarchical Clustering

In the following example we use the data from the last part to plot the hierarchical clustering dendrogram using complete, single, and average linkage clustering, with Euclidean distance as the dissimilarity measure.

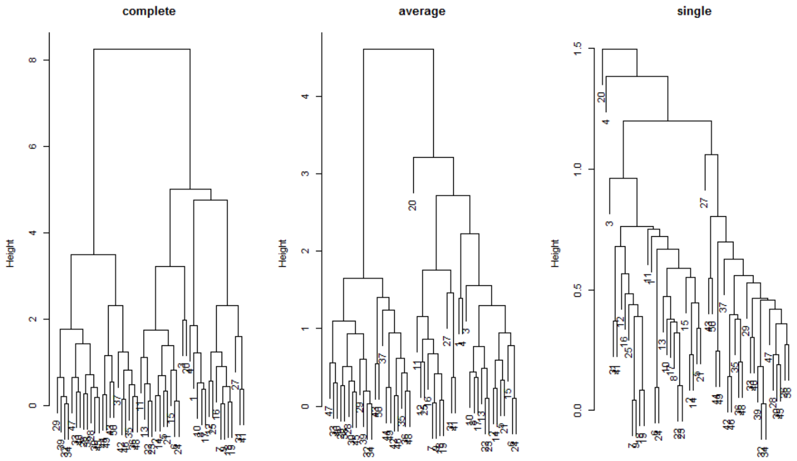
The dist() function is used to compute the pairwise inter-observation Euclidean/-maximum/manhattan distance matrix.

The hclust() function implements hierarchical clustering in R.

```
> d_euc = dist(x, method = "euclidean")
> hc_complete = hclust(d = d_euc, method = "complete")
> hc_average = hclust(d = d_euc, method = "average")
> hc_single = hclust(d = d_euc, method = "single")
```

We can now plot the dendrograms obtained using the usual plot() function. The numbers at the bottom of the plot identify each observation.

```
> par(mfrow = c(1, 3))
> plot(hc_complete, main = "complete")
> plot(hc_average, main = "average")
> plot(hc_single, main = "single")
```

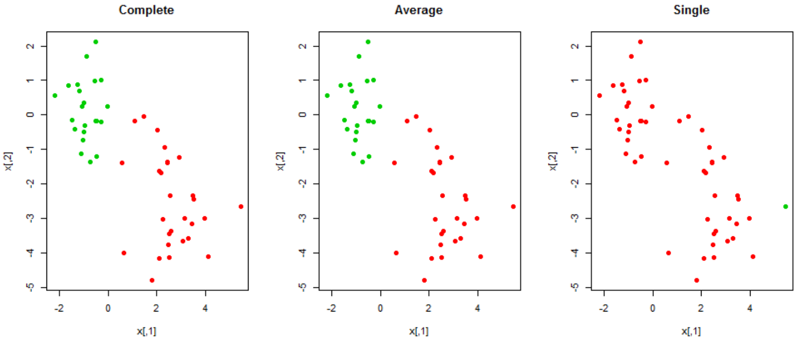


Clustering - Hierarchical Clustering

To determine the cluster labels for each observation associated with a given cut of the dendrogram, we can use the cutree() function.

```
> par(mfrow = c(1,3), pch = 19)
> hcCutComplete = cutree(hc_complete, k = 2)
> plot(x, col = hcCutComplete + 1, main = "Complete")
> hcCutAverage = cutree(hc_average, k = 2)
> plot(x, col = hcCutAverage + 1, main = "Average")
> hcCutSingle = cutree(hc_single, k = 2)
> plot(x, col = hcCutSingle + 1, main = "Single")
```

Clustering - Hierarchical Clustering



For this data, The complete/average linkage generally separate the observations into their correct groups. However, single linkage identifies one point as belonging to its own cluster.

Clustering - Hierarchical Clustering

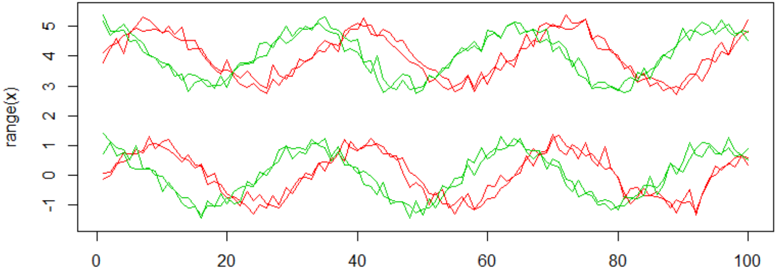
Let's look at the following example. Assumes that we have 50 "curve", first 25 observations are generated from Sine curves and last 25 are Cosine. And No. $1, 3, 5, \dots, 49$ are fluctuating around 4 while No. $2, 4, \dots, 50$ are around 0.

```
> set.seed(1234)
> x0 = seq(0, 2*pi, length.out = 100)
> x = matrix(0, nrow = 50, ncol = 100)
> for(i in 1:50){
+   if(i <=25){
+     x[i,] = sin(x0*pi) + (i%%2)*4 + rnorm(100, sd = 0.2)
+   }else{
+     x[i,] = cos(x0*pi) + (i%%2)*4 + rnorm(100, sd = 0.2)
+   }
+ }
```

Clustering - Hierarchical Clustering

```
> plot(x=c(1, 100), y=range(x), type="n", xlab="", ylab ="")
> lines(x[1,], col = 2);lines(x[2,], col = 2)
> lines(x[3,], col = 2);lines(x[4,], col = 2)
> lines(x[26,], col = 3);lines(x[27,], col = 3)
> lines(x[28,], col = 3);lines(x[29,], col = 3)
```

Distance: “mean” or “pattern” ?



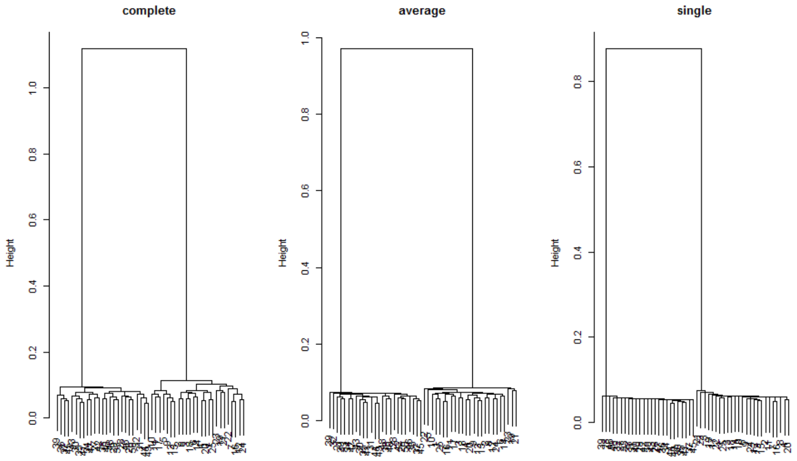
Clustering - Hierarchical Clustering

Correlation-based distance can be computed using the as.dist() function, which converts an arbitrary square symmetric matrix into a form that the hclust() function recognizes as a distance matrix.

```
> correlation = cor(t(x))
> dist_cor = 1 - correlation
> dist_cor = as.dist(dist_cor)
>
> hc_complete = hclust(d = dist_cor, method = "complete")
> hc_average = hclust(d = dist_cor, method = "average")
> hc_single = hclust(d = dist_cor, method = "single")
> cutree(tree = hc_average, k = 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 2 2 2 2 2 2 2
[33] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

Clustering - Hierarchical Clustering

```
> par(mfrow = c(1, 3))
> plot(hc_complete, main = "complete")
> plot(hc_average, main = "average")
> plot(hc_single, main = "single")
```


complete
average
single
Height
Height
Height