To find a single partition into K clusters of a set of N objects in a p dimensional space. Two types of criteria are commonly found:

- Global criterion such as to represent each cluster by a *type-object* (e.g., centroid, medoid) and to assign each object to the nearest *type-object*, optimizing some global criterion of internal homogeneity and/or external heterogeneity, such as, minimizing the within cluster inertia

  Usually requires a prior estimate of the number of clusters

  Examples: k-means and k-medoids (PAM) algorithms
- Local criterion such as to seek for regions of higher density in data. May require to set some parameters

  Example: **DBSCAN**

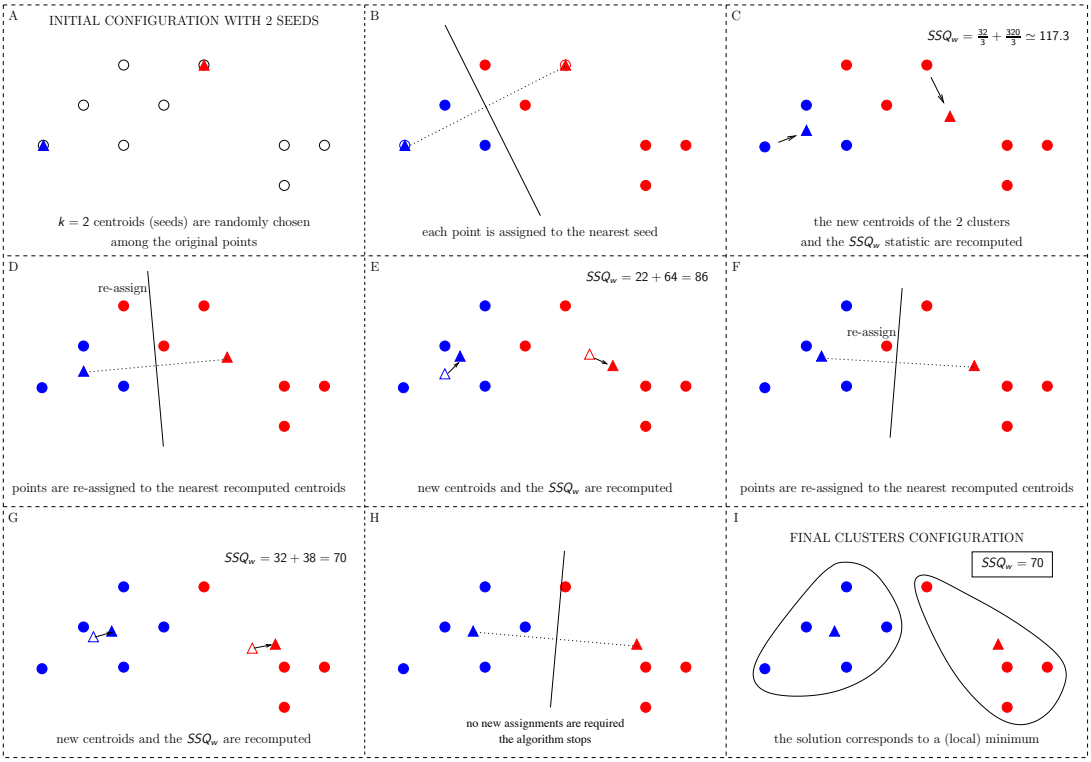
Shares the same global criterion with Ward's method:

To minimize the total within-clusters sum of squares (SSQ_w) of a set of points partitioned into K clusters in a d-dimensional space

Algorithm (MacQueen)

1. *Starts with K randomly chosen initial seeds representing initial candidates to centroids;*
2. *Assigns each object to the nearest centroid*
3. *Recomputes the centroids of the K groups and use them as the new seeds*
4. *Repeat the steps 2 and 3 until no new reassignments occur (in pratice, until the differences between the old seeds and the new recomputed seeds are below a given tolerance threshold)*

k-means algorithm



Convergence of the k-means algorithm

The k-means algorithm consists essentially of a sequence of two steps that are repeatedly iterated:

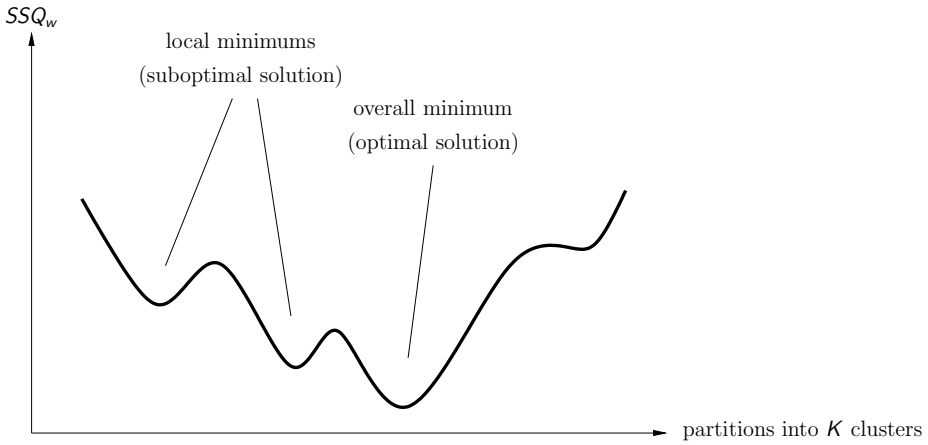
- Reassignment of the points of X to the closest centroid - this step clearly lowers the statistic $SSQ_w = \sum_{k=1}^{K} \sum_{x \in C_k} \|x - m_k\|^2$
- Recalculation of the centroids of the K groups to use as the new seeds - this step also lowers the SSQ_w statistic, since it is a well known fact that the minimum of the quadratic function

$$f(y) = \sum_{x \in G} \|x - y\|^2,$$

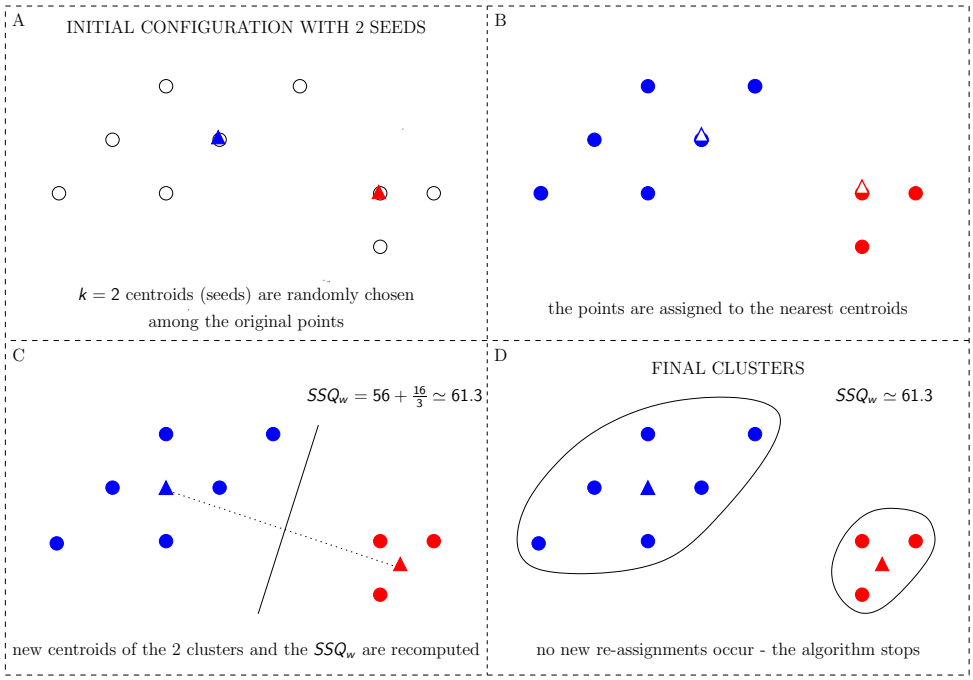
with G a finite subset of $\mathbb{R}^d$, is attained at the centroid of G, i.e., when $y = m_G$

Since there are only finite number of partitions of X into K clusters, the algorithm cannot continue indefinitely strictly lowering the SSQ_w statistic and therefore has to converge to a (possibly local) minimum

The clustering solution can be highly depend on the choice of the initial position of the centroids (seeds) and may converge to a local minimum



The solution found by the k-means algorithm in the previous example is not a global minimum. Actually, with new seeds the algorithm can converge to a solution that improves (i.e., lowers) the SSQ_w statistic



- To repeat the algorithm several times with randomized sets of K seed points and keep the configuration giving the smallest SSQ_w value of the within-cluster inertia
- To provide an initial configuration of K seed points close to the final solution relying on some real hypothesis
- To provide an initial configuration of seed points issued from some hierarchical aggregation method (e.g., Ward), using for instance, their clusters centroids - this is sometimes called the consolidation of the hierarchical clustering

Given a set of N points in the plane,

$$\{c_1, \ldots, c_K\}$$

the Voronoi diagram is defined as the partition of the plane into K convex regions, called Voronoi cells,

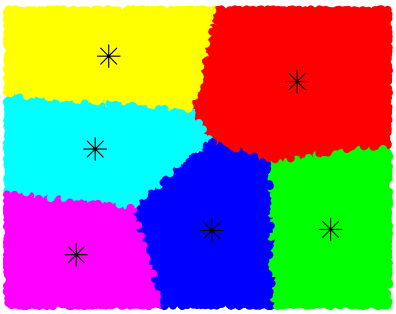
$$R_1, \ldots, R_K$$

such that each cell R_i consists of the set points of the plane closest to c_i

In each step of the k-means algorithm each cluster corresponds to the set of points of X belonging to one of the Voronoi cells defined be the K centroids $c_1, \ldots, c_K$

The above construction can generalized to a set of K points in the N-dimensional space

The partition below into 6 clusters was obtained applying the k-means algorithm to a highly dense set of points in the plane with 6 seeds, to give an approximated idea of the Voronoi cells defined by the final centroids



Each cluster arising from a k-means clustering algorithm lies inside the Voronoi cell containing the respective cluster centroid.
In particular, the convex hulls of the clusters don't overlap, i.e., each pair of clusters can be linearly separated.

The k-means clustering can be performed using the R function

kmeans(x, centers, iter.max = 10, nstart = 1, ...)

x: numeric matrix of data

centers: the number of clusters or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in x is chosen as the initial centres

nstart: if centers is a number, how many random sets should be chosen (repeat)
Returns a list with components:

cluster: A vector of integers (from 1:k) indicating the number of the cluster where each point is assigned

centers: A matrix of cluster centers.

totss: The total sum of squares, i.e., SSQ_t

withinss: Vector of within-cluster sum of squares, one component per cluster

tot.withinss: Total within-cluster sum of squares, i.e., SSQ_w

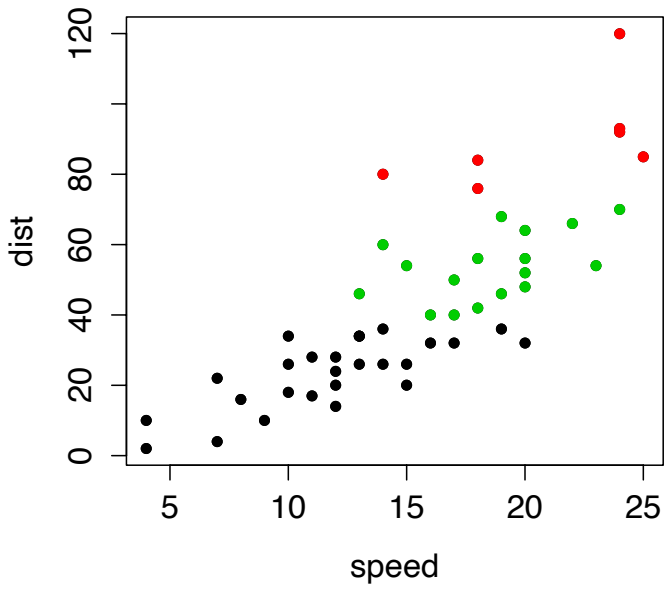
betweenss:The between-cluster sum of squares, i.e., SSQ_b

size: The number of points in each cluster

R

```
require(datasets)
data(cars)
?cars
head(cars)
cars.cl<-kmeans(cars, 3, nstart=100)
# 3 centers randomly chosen repeated 100 times
cars.cl
plot(cars,type=‘‘p’’,pch=16,cex=.5)
for(i in 1:50){points(cars[i,1],
cars[i,2],col=cars.cl$cluster[i], pch=16,type=‘‘p’’)}
```


Clustering result



- The optimizing function SSQ_w is always monotonic decreasing, i.e., the intra-group inertia decreases in each step, converging to some (possibly local) optimum
- The number of iterations required to converge is usually small (≈ 10 iterations are enough)
- Finding an optimal solution is NP-hard. Actually the time complexity is $O(n^{dK+1} \ln d)$, where K denotes the number of clusters, d the dimension and N the number of points)
- It tends to form rounded shaped clusters that can be linearly separated (since each cluster is contained in a Voronoi cell).

  In particular, it cannot detect arbitrarily shaped clusters
- Nearby points can end in distinct classes. Groups can end empty
- Sensitive to noise and outliers
- Requires some geometric notion of centroid. In particular, it cannot be applied to categorical data assumes that the points lie in some euclidean space

The model-based clustering as a generalization of k-means

- The standard model-based clustering is a finite mixture of multivariate Gaussians, i.e., it is assumed that each cluster C_i is generated by a multivariate Gaussian distribution with pdf

$$\phi(x|\mu_i, \Sigma_i)$$

  where μ_i and Σ_i are the mean and covariance matrix of C_i
- One seeks a partition of X into clusters C_i and a mixture of Gaussians with pdf given by a convex combination of the form

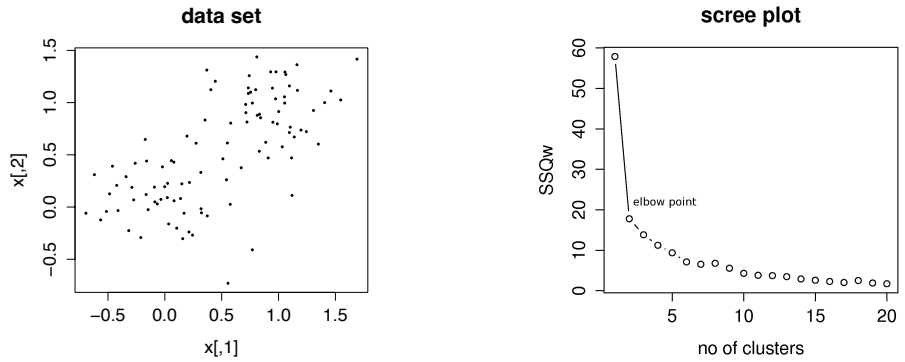
$$\phi = \sum_i \eta_i \phi(x|\mu_i, \Sigma_i),$$

  with nonnegative weights η_i, $i = 1, \ldots, K$, such that $\sum_i \eta_i = 1$. To determine the parameters uses the so-called **expectation-maximization** algorithm
- In the model-based clustering the partition can have clusters with different covariance matrices i.e., with distinct ellipsoidal shapes, volumes and orientations, that account with distinct weights to the pdf of the finite mixture
- The k-means clustering can be considered a particular case of the model-based clustering, with all weights η_i equal to $\frac{1}{K}$ and identical isotropic covariance matrices $\Sigma_i = \sigma^2 I$ (I denotes the identity matrix).

- To estimate the optimal number of clusters we usually look for a good trade-off between a relatively small number of clusters (parsimony principle) and the minimization of the information (variability) loss due to replacing the observations in each cluster by some cluster representative (for instance, the cluster centroid).
- This is one of the most difficult tasks in clustering analysis and no definitive answer can usually be given.
- Several internal cluster validity indices can be used to estimate the optimal number of clusters and/or to assess the cluster quality. Among the most well-known indices we have:
  - SSQ_w.
  - Calinski-Harabasz index.
  - Silhouette coefficient.
  - Davies-Boudin.
  - Duhn index.
  - Several other indices can be computed with the R functions `clustCrit` and `NbClust`.

For a more detailed account on validity indices, See, for instance, O. Arbelaitz et al. *An extensive comparative study of cluster validity indices*, Pattern Recognition 46 (2013) 243–256

- A simple method to estimate the best number of clusters consists to study the variation of SSQ_w with number of clusters in a scree plot, which essentially amounts, by Huygens's theorem, to study the variation of the percentage of total inertia retained by the clusters, i.e., explained by the partition, $\frac{SSQ_b}{SSQ_t}$
- An elbow point in the scree plot indicating high decrease in the SSQ_w statistic while further increments in the number of clusters will only marginally improves this statistic, could suggest a good estimate for the optimal number of clusters



- Although the statistic SSQ_w depends on the number of clusters, it can be used to compare partitions of a given dataset X with the same number of clusters. Partitions yielding smaller SSQ_w values are preferable for this criterion.

- The Calinski-Harabaz index also known as variance ratio criterion is defined as

$$CH(K) = \frac{SSQ_b/(K-1)}{SSQ_w/(N-K)}$$

with the optimal number of clusters being estimated as the number yielding the largest value for $CH(K)$. (Inspired in the F-ratio test of one-way ANOVA)

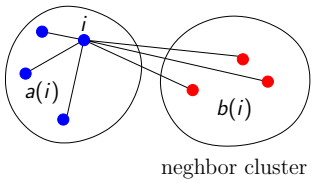
- Since we have

$$\begin{aligned} CH(K) &= \frac{SSQ_b/(K-1)}{SSQ_w/(N-K)} = \frac{N-K}{K-1} \times \frac{SSQ_b}{SSQ_w} \\ &= \frac{N-1+1-K}{K-1} \times \frac{SSQ_b}{SSQ_w} = \left(\frac{N-1}{K-1} - 1\right) \frac{SSQ_b}{SSQ_w}, \end{aligned}$$

high values of $CH(K)$ are obtained with well separated and homogeneous clusters, i.e., with large values of SSQ_b and small values of SSQ_w, keeping at the same time, the number of clusters K relatively small, i.e., $\frac{N-1}{K-1}$ relatively large.

- Particularly well adapted when clusters tend to have spherical shapes due to its definition based on the variance
- Several studies suggest Calinski-Harabaz index as being one of the internal cluster validity indices yielding the best results - see, for instance one of the reference papers on internal cluster validation,
Milligan GW, Cooper MC (1985) An Examination of Procedures for Determining the Number of Clusters in a Data Set. Psychometrika 50:159–179.
- Can be computed using the R function `calinhara` of the package `fpc`

- For each observation i we compute the average dissimilarity $a(i)$ between i and the remaining points in its cluster
- For each one of the other clusters we compute the average dissimilarity from point i to the points of that cluster and take the minimum $b(i)$ of these average dissimilarities
- The cluster for which the minimum $b(i)$ is attained, i.e., the cluster with lowest average dissimilarity w.r.t to observation i, is called the **neighbor cluster** of i

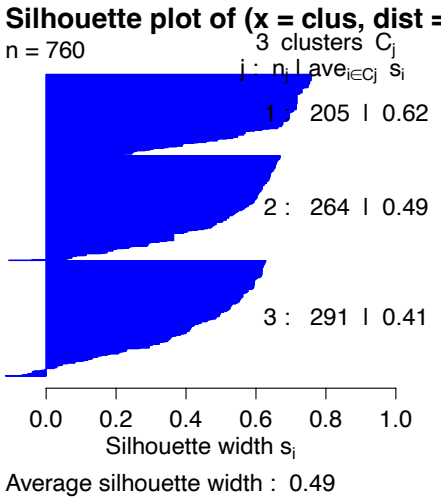


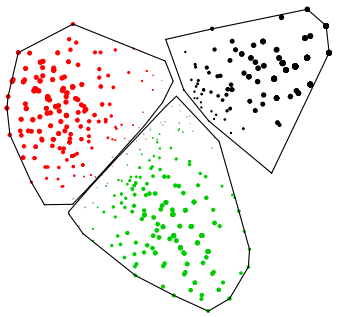
The silhouette coefficient of observation i is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

and gives an indication of how well an element is classified in its cluster

- The denominator $\max\{a(i), b(i)\}$ is a normalization term allowing that the index vary in the range $[-1, 1]$
- Small values of $a(i)$ along with large values of $b(i)$ yield a silhouette coefficient close to one
- Likewise, large values of $a(i)$ along with small values of $b(i)$ yield a silhouette coefficient close to minus one
- Observations with silhouette coefficients close to one are very well classified
- Observations with silhouette coefficients close to zero probably lie between clusters
- Observations with negative silhouette coefficients are probably misplaced in their clusters





In the figure on the right the dot sizes are proportional to their silhouette coefficients. Larger dots lie in core regions of the clusters whereas smaller dots lie in border regions or between clusters

Average silhouette width - an internal validity criterion

The average silhouette width (ASW) is defined as the average of the silhouette coefficients for all observations

- It assess both cluster cohesion and cluster separation
- It increases with a strong cluster separation (higher $b(i)$ values) and cluster tightness (small values of $a(i)$)

Range of ASW

It is common to consider that

- between 0.71 and 1.0: a strong structure has been found
- between 0.5 and 0.7: a reasonable structure has been found
- between 0.26 and 0.5: the structure is weak and can be artificial
- below 0.25: no substantial structure has been found

The optimal number of clusters can be estimated maximizing the ASW

A closely related internal validation criterion is Davies-Bouldin index

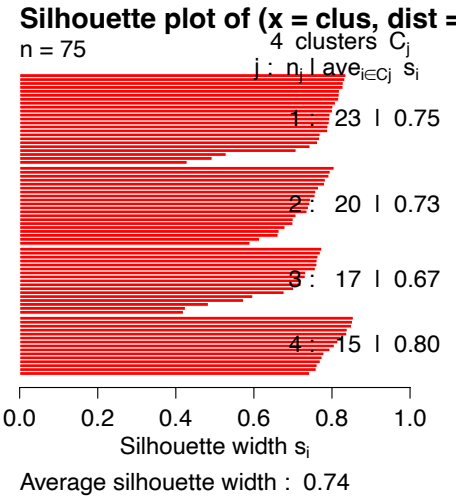
$$DB = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\frac{S_i + S_j}{m_{ij}}$$

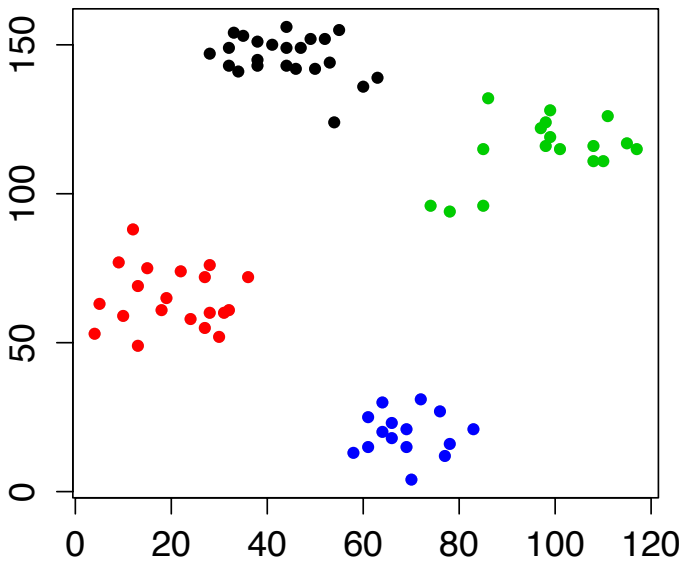
Here S_i denotes some internal cohesion measure of cluster C_i and m_{ij} a separation measure between clusters C_i and C_j, verifying certain properties...
For instance, S_i can be the average distance of the points of C_i to its centroid and m_{ij} the distance between the centroids of C_i and C_j

Number of clusters?

Applying the criteria SSQ_W statistic, ASW and CH to the Ruspini data, a popular dataset in clustering analysis, all criteria agree on 4 clusters

The average of the silhouette widths of the previous example is close to .75 suggesting that a strong clustering structure was found in Ruspini data. Since all silhouette coefficients are above .4 no points are misplaced in their clusters





Ruspini plot into 4 clusters using the k-means algorithm

R (code)

```
library(cluster)
ch.res<-rep(NA,10)
si.res<-rep(NA,10)
ssqw.res<-rep(NA,10)
plot(ruspini)
for (n in 2:10){
km <- kmeans(ruspini,n,nstart=500)
ch.res[n]<-round(calinhara(ruspini,km$cluster),digits=2)
si.res[n]<-mean(silhouette(km$cluster,dist(ruspini))[,3])
ssqw.res[n]<-km$tot.withinss
# ssqw.res[n]<-km$betweenss/km$tot.withinss
}
par(mfrow=c(2,2))
plot(ssqw.res,type="b",col="black",main="SSQw")
plot(si.res,type="b",col="blue",main="SIL")
plot(ch.res,type="b",col="red",main="CH")
km <- kmeans(ruspini,4,nstart=500)
plot(ruspini, col=km$cluster)
```


COMPARING PARTITIONS

- Several clustering analyses of the same data can be done using distinct meaningful combinations of clustering methods and resemblance notions;
- Clustering analyses having a high degree of agreement may suggest that the common patterns produced by these methods is robust;
- If the clustering structure is known *a priori* and it is important to assess how well the clustering method was able to reproduce this structure;
- It is very difficult (if not impossible or meaningless) to match each cluster of a partition with the correct cluster of the other partition
- The usual way is to compute the number of pairs of individuals that both clustering methods agree to assign in the same/distinct class

- Assume that N individuals are classified by two distinct clustering methods. The total number of pairs of individuals is $\binom{N}{2} = \frac{N(N-1)}{2}$. Denote by:

  A: number of pairs classified in the same class in both partitions

  B: number of pairs classified in the same [distinct] class in the first [second] partition

  C: number of pairs classified in the distinct [same] class in the first [second] partition

  D: number of pairs classified in distinct classes in both partitions

- The above quantities can be represented in a contingency table as follows:

	Part. 2		
Part. 1	Classif. in the same group	Classif. in distinct groups	
Classif. in the same group	A	B	A+B
Classif. in distinct groups	C	D	C+D
	A+C	B+D	$\binom{N}{2}$

- Rand index (RI) is a simple concordance index used as an external validity index to compare partitions and is defined as,

$$RI = \frac{A+D}{\binom{N}{2}} = \frac{A+D}{A+B+C+D},$$

  where $A+D$ is the number of agreements for both partitions
- It ranges from 0 *(total disagreement)* to 1 *(total agreement)*
- To each partition of a set of N individuals, $x_1, \ldots, x_N$ we associate a binary vector of length $\binom{N}{2}$, where the component corresponding to pair (i,j) is equal 1 if x_i and x_j are assigned in the same class and 0 otherwise
- The Rand index of two partitions is just the simple matching index between the binary vectors associated to these partitions
- Note that the number of groups in each partition can be distinct

$X = \{a, b, c, d, e, f\}$

Partition 1: $a\ b\ e \mid c \mid d\ f$ Partition 2: $a\ c \mid b\ d \mid e\ f$

	a	b	c	d	e
b	1	·	·	·	·
c	0	0	·	·	·
d	0	0	0	·	·
e	1	1	0	0	·
f	0	0	0	1	0

	a	b	c	d	e
b	0	·	·	·	·
c	1	0	·	·	·
d	0	1	0	·	·
e	0	0	0	0	·
f	0	0	0	1	1

The contingency table between partition 1 and partition 2 is

	1	0	
1	A	B	$A+B$
0	C	D	$C+D$
	$A+C$	$B+D$	$\binom{N}{2}$

$=$

	1	0	
1	**0**	4	4
0	3	**8**	11
	3	12	**15**

Hence

$$RI = \frac{\mathbf{0}+\mathbf{8}}{\mathbf{15}} = 0.53333\ldots$$

Computing the Rand index in R

To compute the Rand index of the two partitions in 3 classes,

$$\mathcal{P}_1: \quad a\ b\ e \mid c \mid d\ f \qquad\qquad \mathcal{P}_1: \quad a\ c \mid b\ d \mid e\ f,$$

we encoded these partitions as vectors

$$(1,1,2,3,1,3), \qquad (1,2,1,2,3,3),$$

representing the classes of the elements a, b, c, d, e, f

R (Rand index)

```
# Codigo da funcao do Professor Cadima
rand <- function(class1,class2){
n <- length(class1)
c <- as.dist(outer(class1,class1,"=="))
d <- as.dist(outer(class2,class2,"=="))
rand <- sum(c == d)/(n*(n-1)/2)
return(rand) }
rand(c(1,1,2,3,1,3),c(1,2,1,2,3,3))
# 0.5333333
2 random samples of length 1000 with elements extracted from 1,...,10
p1<-sample(1:10,1000,replace=TRUE)
p2<-sample(1:10,1000,replace=TRUE)
rand(p1,p2)
# 0.8196997
```


Correction for chance: adjusted Rand index

The expected value of Rand index between random partitions is not constant (e.g., 0). To overcome this issue Hubert and Arabie proposed the so-called adjusted Rand index

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} = \frac{RI - E[RI]}{1 - E[RI]},$$

assuming the Permutation Model as the null model for random clusterings i.e., each partition $\mathcal{P}_i$, $i = 1, 2$, is drawn at random, subject to having a prescribed number of classes K_i and a prescribed number of elements $N_{i,j}$ in each class $j = 1, \ldots, K_i$.

It can be proved that,

$$E[RI] = \frac{2Q_1\,Q_2 - \binom{N}{2}(Q_1 + Q_2) + \binom{N}{2}}{\binom{N}{2}^2},$$

where $Q_i = \sum_{j=1}^{K_i} \binom{N_{ij}}{2}$, $i = 1, 2$, yielding

$$ARI = \frac{\binom{N}{2}(A + D) - U}{\binom{N}{2}^2 - U},$$

where $U = (A + B)(A + C) + (D + B)(D + C)$ and $\binom{N}{2} = \frac{N(N-1)}{2}$.

$ARI \in [-1, 1]$ with $ARI \approx 0$ for independent random partitions, $ARI = 1$ for identical partitions and $ARI < 0$ if the partitions have a low agreement.

More difficult to interpret than the more simple Rand index