Unsupervised Method - Clustering

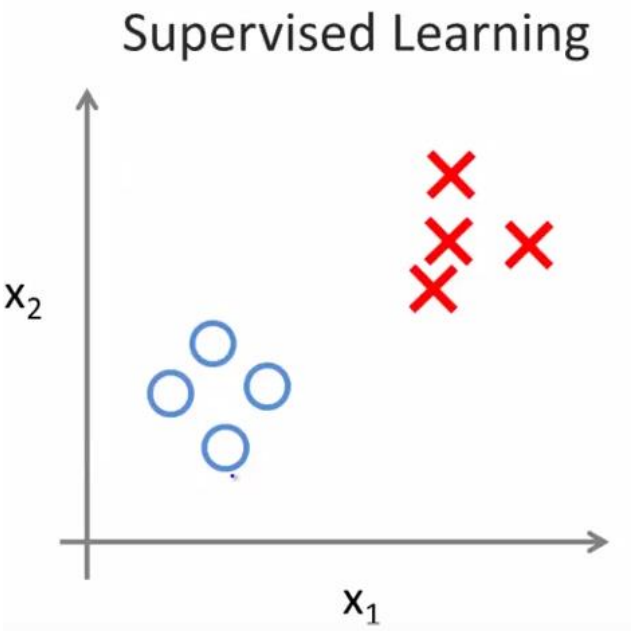
Data Science

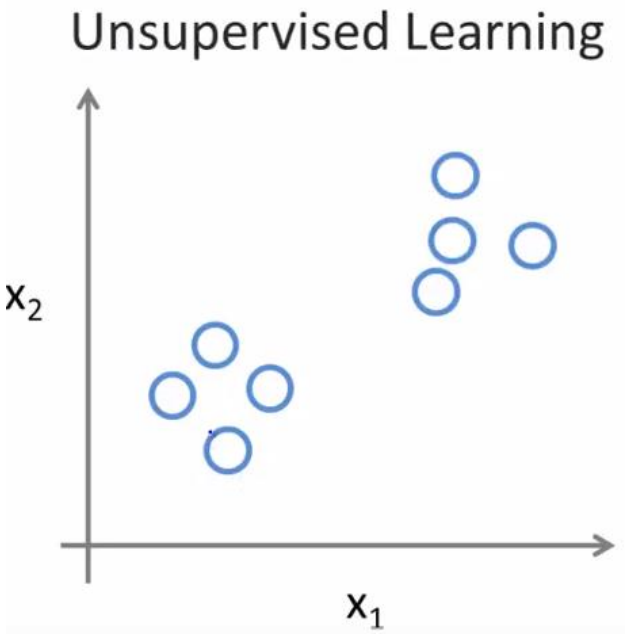
Presented by Hyebong Choi

Unsupervised Method

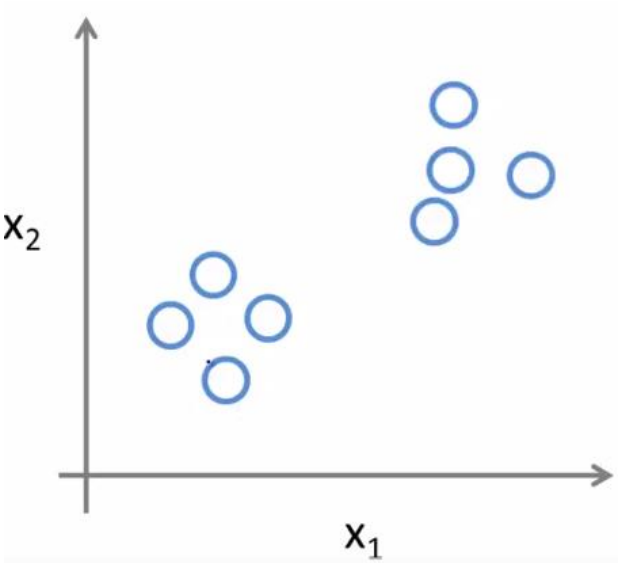
This topic corresponds to chapter 8 of the text book

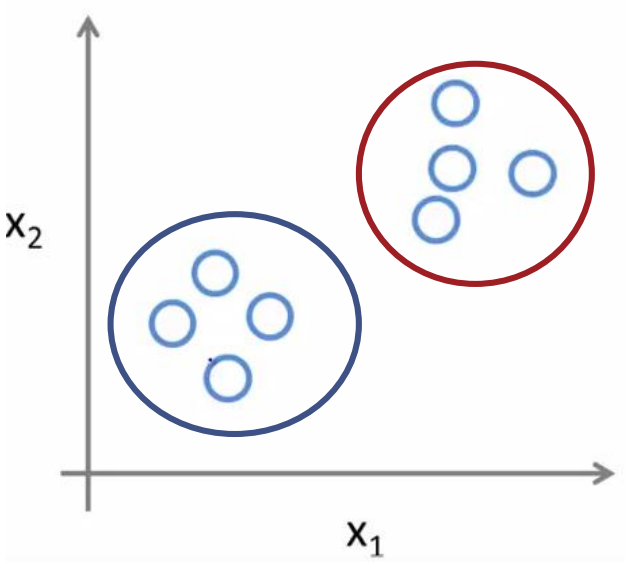
UNSUPERVISED LEARNING





CLUSTERING



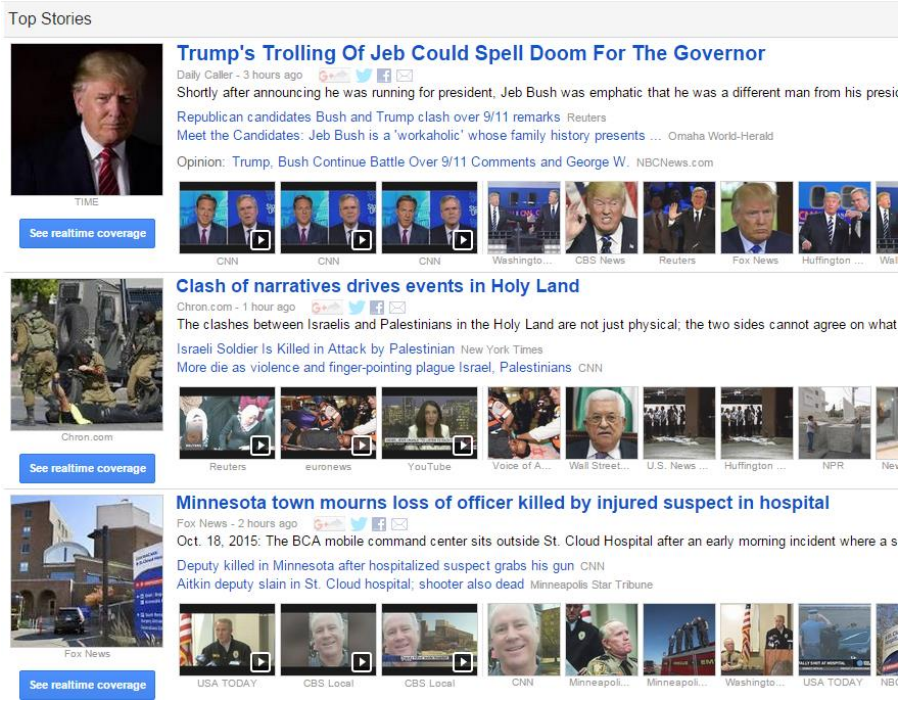


WHY UNSUPERVISED LEARNING?

- ✓ Labeled data are relatively rare compared to unlabeled ones
- ✓ To find latent structure from features alone
- ✓ Can be goal in itself
  - ✓ discover hidden patterns, exploratory data analysis
- ✓ Can be means to an end
  - ✓ preprocessing for supervised task, learning features

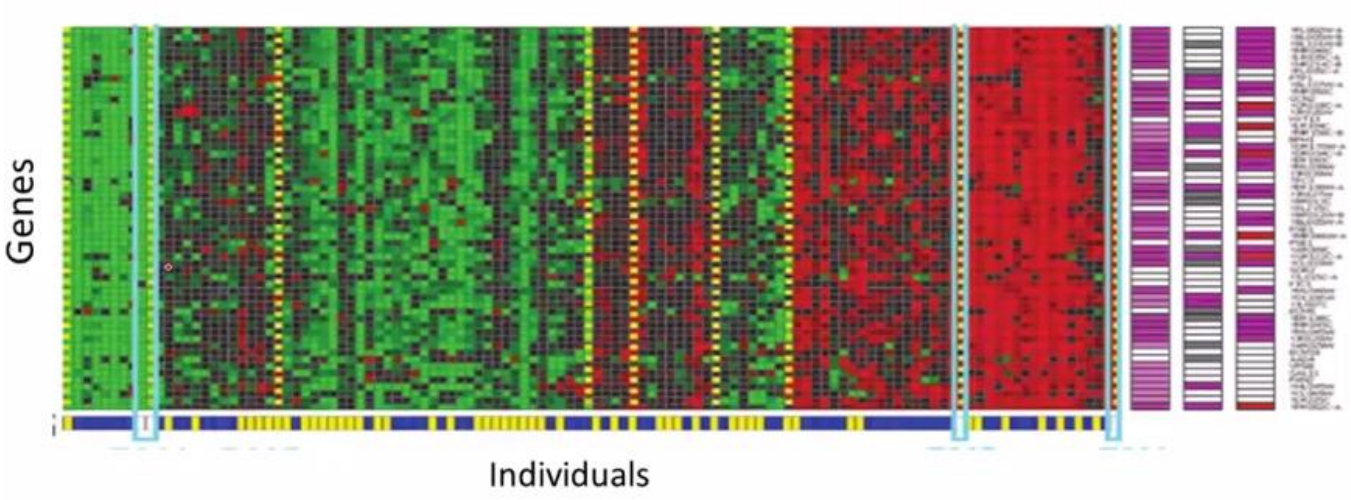
CLUSTERING EXAMPLE

- ✓ **Google new clustering (news.google.com)**



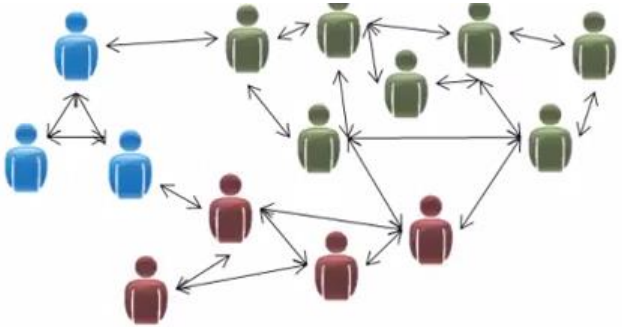
CLUSTERING EXAMPLE

- ✓ Genome micro-array



UNSUPERVISED LEARNING EXAMPLES

Organize computing clusters



Social network analysis

Market segmentation

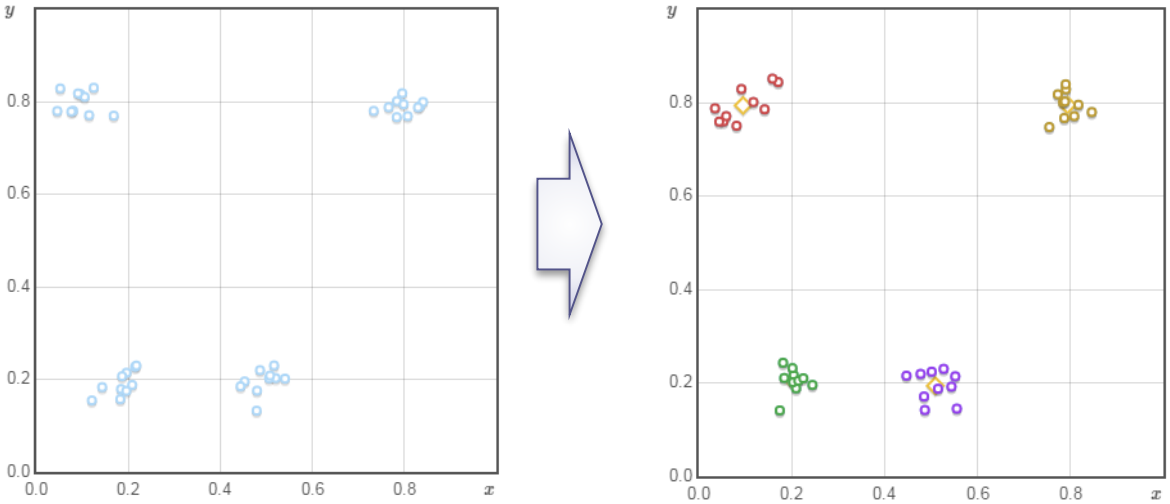
Astronomical data analysis

k-means clustering

1967 by MacQueen, J

K-means Clustering

- ✓ One of the simplest unsupervised learning methods invented in 1967
- ✓ Goal: given n data points, group the data points into k cluster s.t. data points in a cluster are close each other with respect to predefined similarity measure
  - ✓ *e.g. Euclidean distance*



Procedure

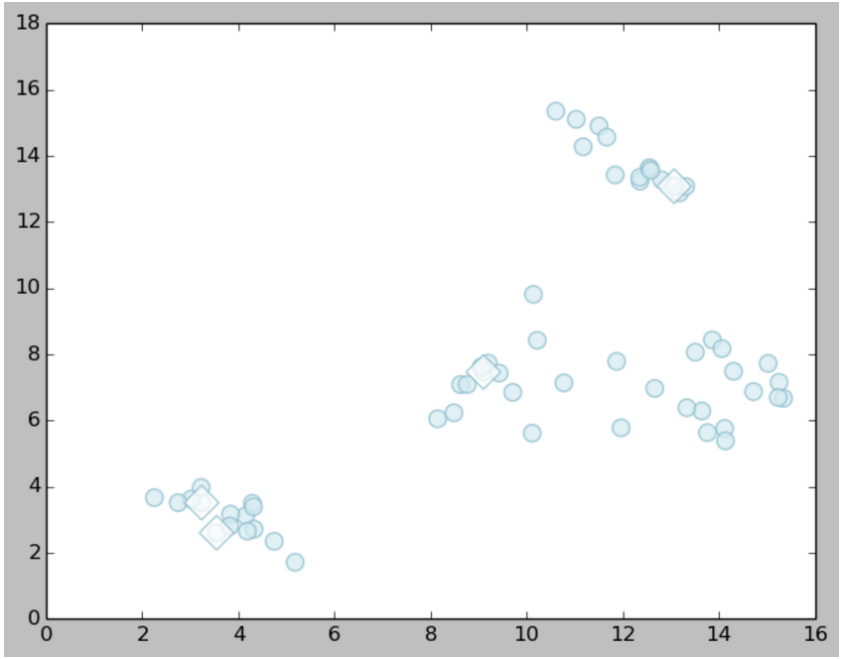
1. Initialize the center of the clusters	$\mu_i =$ some value $, i = 1, \ldots, k$
2. Attribute the closest cluster to each data point	$\mathbf{c}_i = \{j : d(\mathbf{x}_j, \mu_i) \leq d(\mathbf{x}_j, \mu_l), l \neq i, j = 1, \ldots, n\}$
3. Set the position of each cluster to the mean of all data points belonging to that cluster	$\mu_i = \frac{1}{\lvert c_i \rvert} \sum_{j \in c_i} \mathbf{x}_j, \forall i$
4. Repeat steps 2-3 until convergence	
Notation	$\lvert \mathbf{c} \rvert =$ number of elements in $\mathbf{c}$

$$d(\mathbf{x}, \mu_i) = \|\mathbf{x} - \mu_i\|_2^2$$

- Initialization of centroid of clusters: Up to designer's choice
- Forgy: set the positions of the k clusters to k observations chosen randomly from the dataset.
- Random partition: assign a cluster randomly to each observation and compute means of each cluster and set them to centroid.

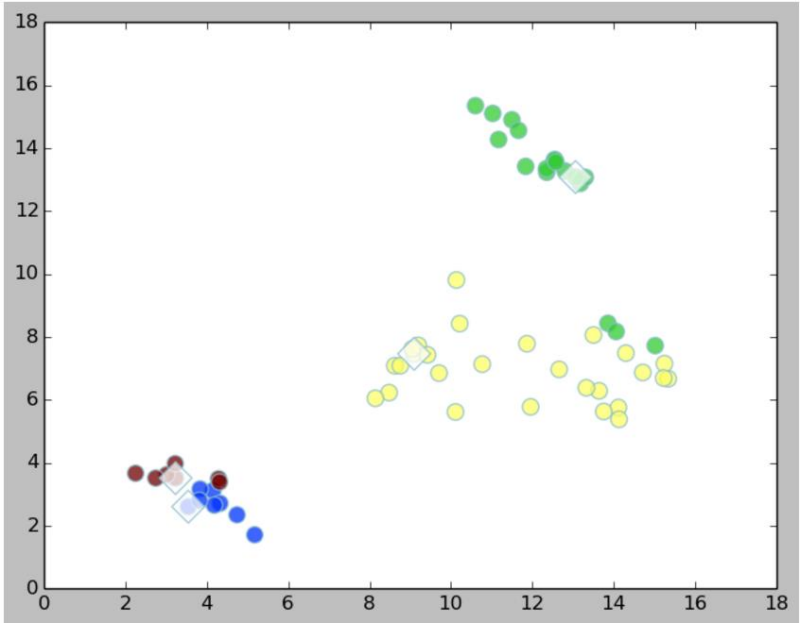
Example

- Select initial centroids: given n data points, select k points randomly



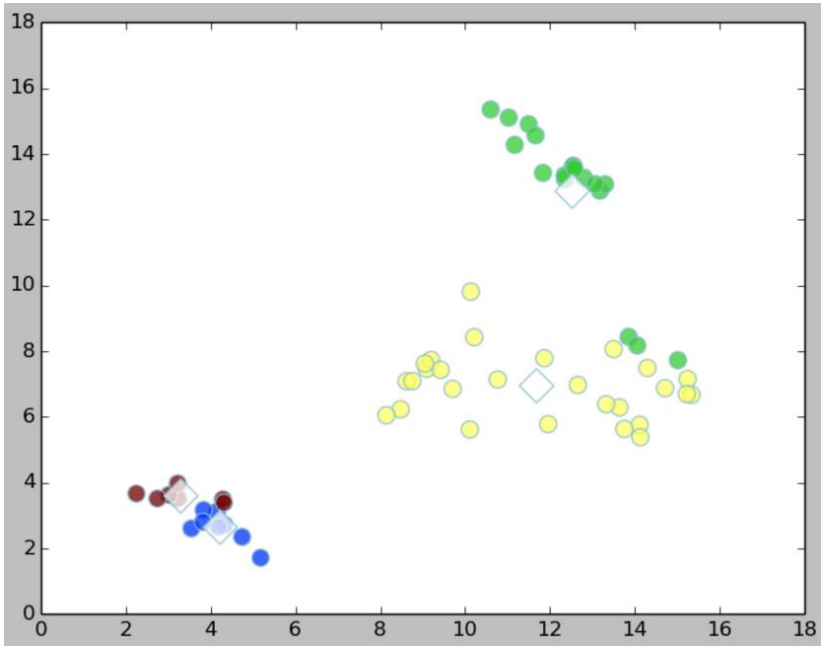
Example

- Assign data points to their closest centroid



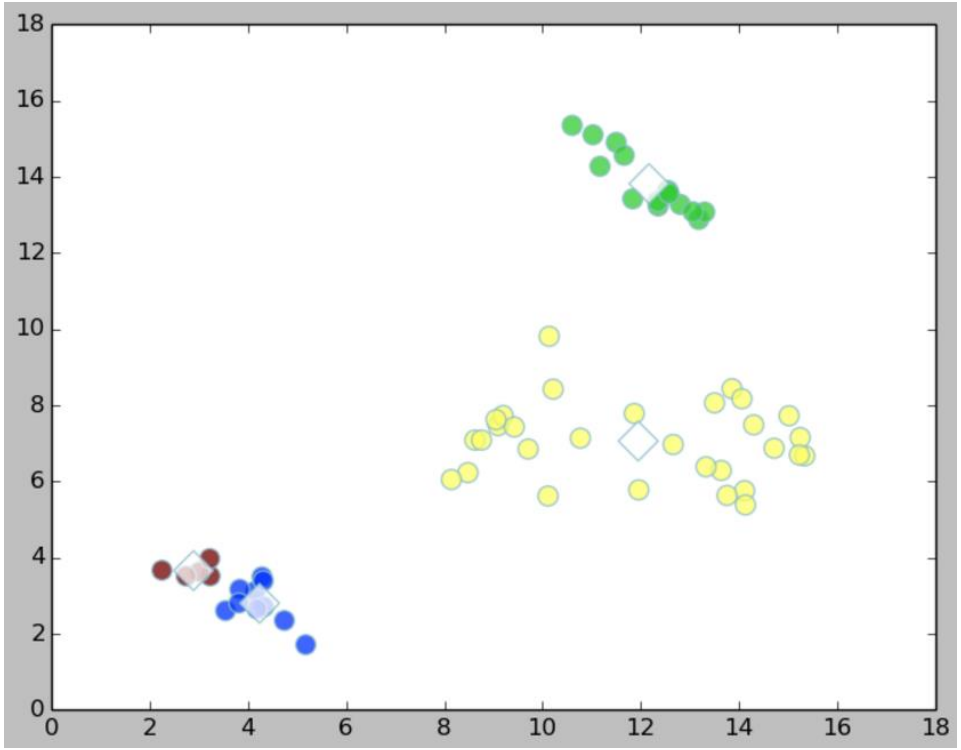
Example

- Re-calculate the centroids as mean of data point in cluster

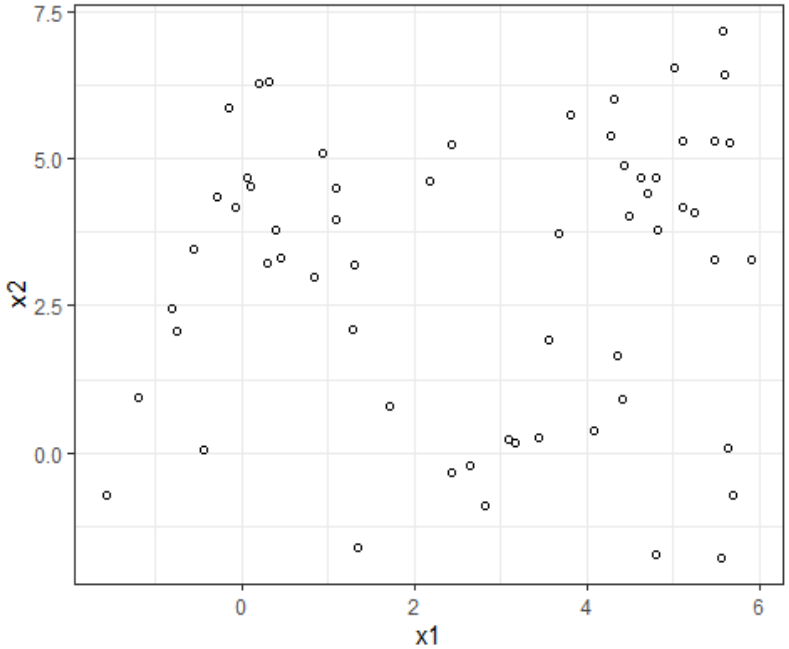


Example

- Repeat steps above until there is no change to clusters

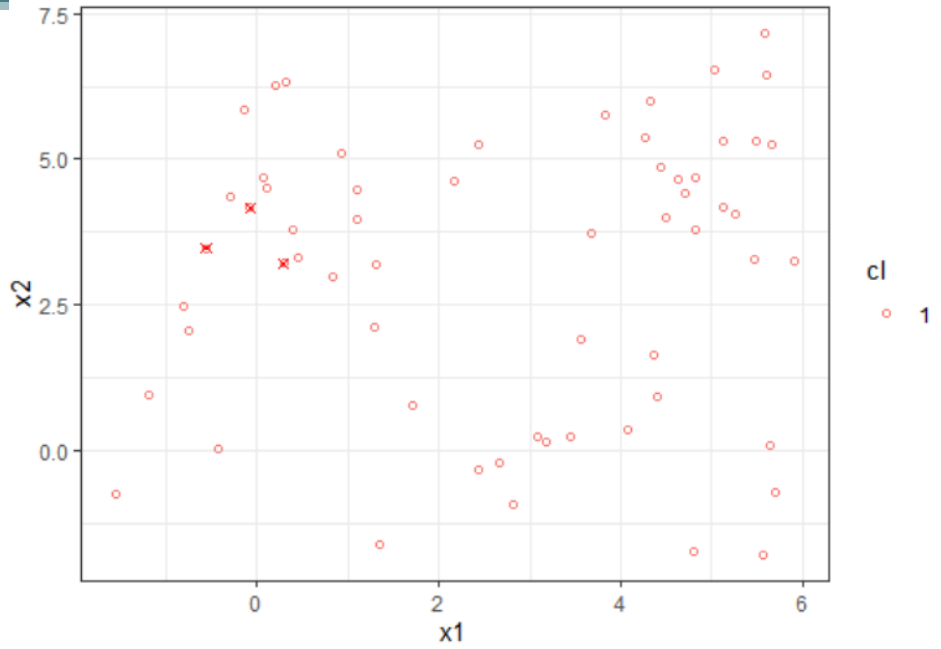


Generating Synthetic Data



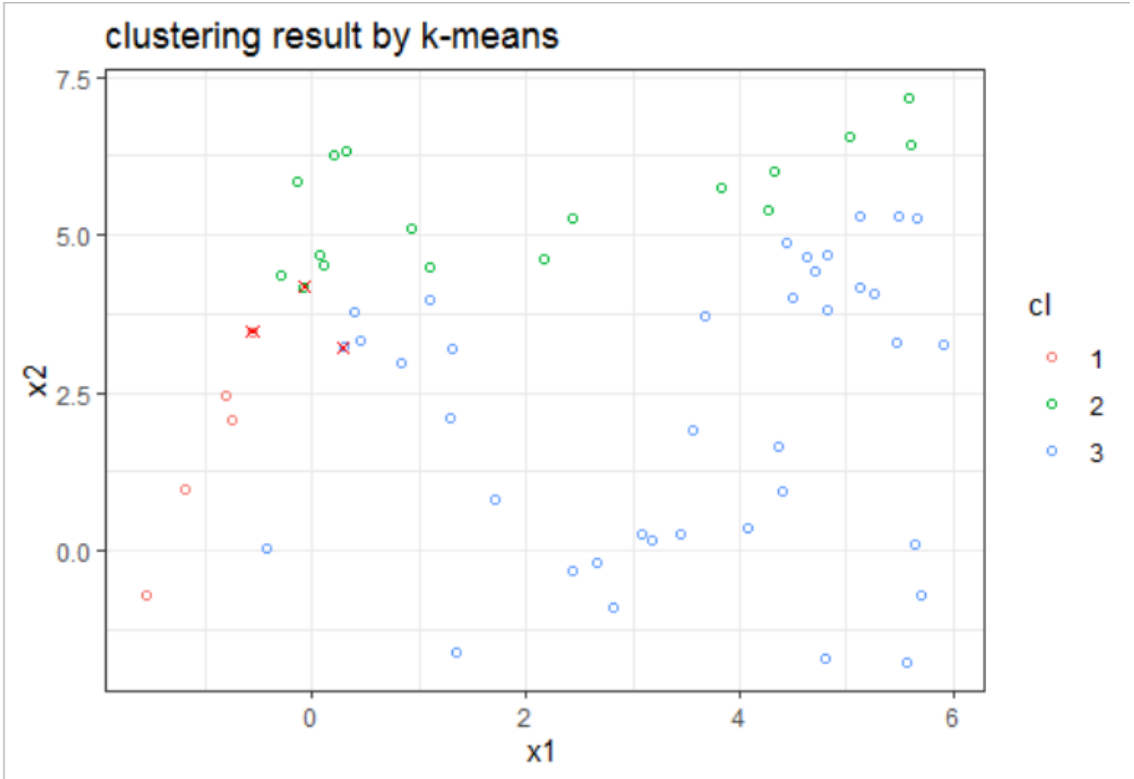

```
set.seed(2020)

synth.data <- data.frame(x1 = c(rnorm(20, 3, 1.5), rnorm(20,0,1), rnorm(20,5,1)),
                         x2 = c(rnorm(20, 0, 1), rnorm(20,4,1), rnorm(20,5,1)))
ndata <- nrow(synth.data)
ndim <- ncol(synth.data)

synth.data %>% ggplot(aes(x = x1, y= x2)) + geom_point(shape = 1) + theme_bw()
```

Initial Setting

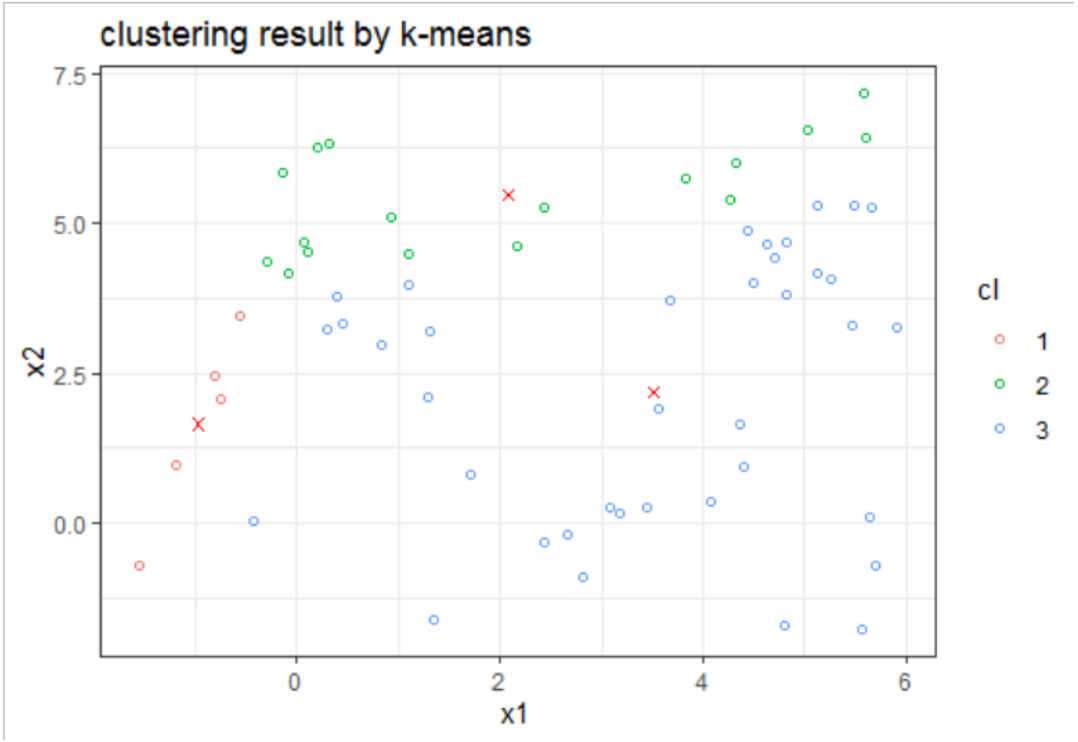



```
u_dist <- function(u, v){
  sqrt(sum((u - v) ** 2))
}

# inital setting

set.seed(2020)
k <- 3
cents <- data.frame(cl = 1:k)
cents <- cbind(cents, synth.data[sample(1:60, k),])


synth.data$cl <- factor(rep(1, ndata), levels = 1:k)
synth.data %>% ggplot(aes(x = x1, y= x2, col = cl)) + geom_point(shape = 1)
+ theme_bw() +
  geom_point(data = cents, shape = 4, col = 'red')
```

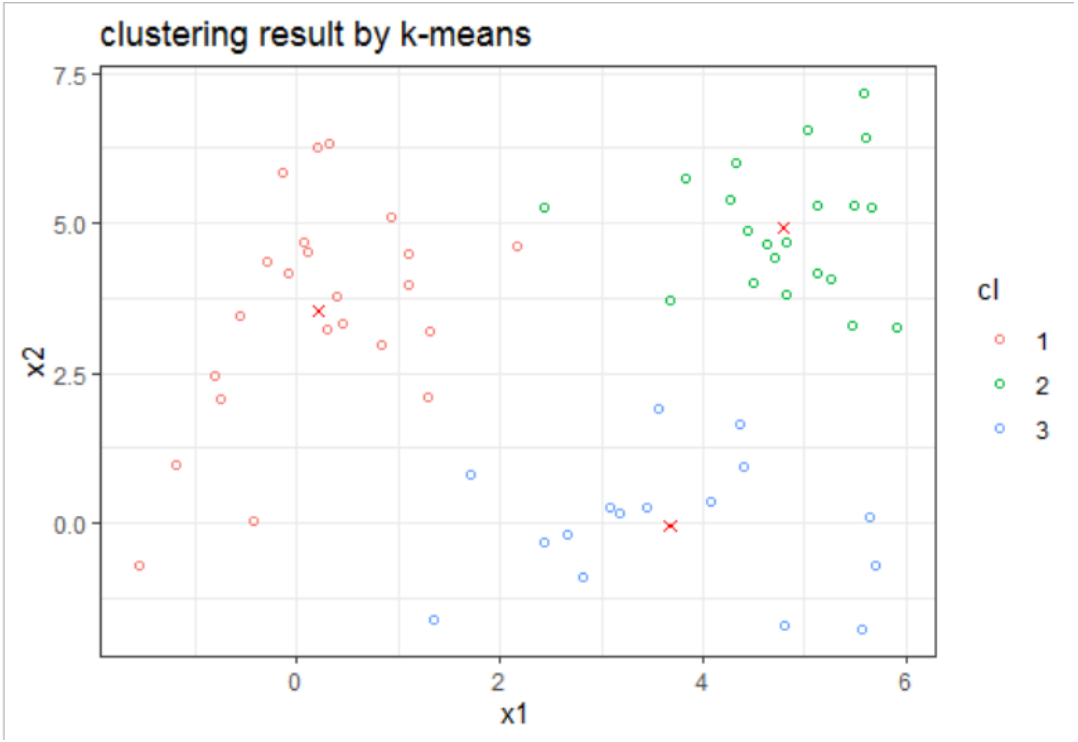
Assign Data Points to their Closest Centroid



Move Centroids to Clusters' Average Point



Repeat Until No Change



Practice

- Implement k-means clustering

```
while(TRUE){
  # data assignment to cluster
  new_cl <-
    apply(synth.data[,1:ndim], 1, function(x) {
      which.min(
        apply(cents[,-1], 1, function(y) {
          u_dist(y, x)
        })
      )
    })

  if(all(synth.data$cl == factor(new_cl))) break

  synth.data$cl <- factor(new_cl)

  cents <- synth.data %>% group_by(cl) %>%
    summarise(x1 = mean(x1), x2 = mean(x2))
}
```

data loading and preparation

```
protein <- read.table("protein.txt", sep="\t", header=TRUE)
summary(protein)
##           Country      RedMeat        WhiteMeat          Eggs
##  Albania       : 1   Min.   : 4.400   Min.   : 1.400   Min.   :0.500
##  Austria       : 1   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.700
##  Belgium       : 1   Median : 9.500   Median : 7.800   Median :2.900
##  Bulgaria      : 1   Mean   : 9.828   Mean   : 7.896   Mean   :2.936
##  Czechoslovakia: 1   3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.700
##  Denmark       : 1   Max.   :18.000   Max.   :14.000   Max.   :4.700
##  (Other)       :19
##       Milk            Fish           Cereals          Starch
##  Min.   : 4.90   Min.   : 0.200   Min.   :18.60   Min.   :0.600
##  1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30   1st Qu.:3.100
##  Median :17.60   Median : 3.400   Median :28.00   Median :4.700
##  Mean   :17.11   Mean   : 4.284   Mean   :32.25   Mean   :4.276
##  3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10   3rd Qu.:5.700
##  Max.   :33.70   Max.   :14.200   Max.   :56.70   Max.   :6.500
##
##       Nuts           Fr.Veg
##  Min.   :0.700   Min.   :1.400
##  1st Qu.:1.500   1st Qu.:2.900
##  Median :2.400   Median :3.800
```

```
vars.to.use <- colnames(protein)[-1]
pmatrix <- scale(protein[,vars.to.use])
pcenter <- attr(pmatrix, "scaled:center")
pscale <- attr(pmatrix, "scaled:scale")
```

protein dataset from 1973 on protein consumption from nine different food groups in 25 countries in Europe.

k-means function

```
pclusters <- kmeans(pmatrix, 5, nstart=100, iter.max=100)
summary(pclusters)
```

```
##              Length Class  Mode
## cluster      25     -none- numeric
## centers      45     -none- numeric
## totss         1     -none- numeric
## withinss      5     -none- numeric
## tot.withinss  1     -none- numeric
## betweenss     1     -none- numeric
## size          5     -none- numeric
## iter          1     -none- numeric
## ifault        1     -none- numeric
```

```
pclusters$centers
```

```
##        RedMeat  WhiteMeat        Eggs       Milk       Fish    Cereals
## 1 -0.508801956 -1.1088009 -0.41248496 -0.8320414  0.9819154  0.1300253
## 2 -0.807569986 -0.8719354 -1.55330561 -1.0783324 -1.0386379  1.7200335
## 3 -0.570049402  0.5803879 -0.08589708 -0.4604938 -0.4537795  0.3181839
## 4  0.006572897 -0.2290150  0.19147892  1.3458748  1.1582546 -0.8722721
## 5  1.011180399  0.7421332  0.94084150  0.5700581 -0.2671539 -0.6877583
##       Starch       Nuts      Fr.Veg
## 1 -0.1842010  1.3108846  1.62924487
## 2 -1.4234267  0.9961313 -0.64360439
## 3  0.7857609 -0.2679180  0.06873983
## 4  0.1676780 -0.9553392 -1.11480485
## 5  0.2288743 -0.5083895  0.02161979
```

```
pclusters$size
```

```
## [1] 4 4 5 4 8
```

```
pclusters$cluster
```

```
##  [1] 2 5 5 2 3 4 3 4 5 1 3 5 1 5 4 3 1 2 1 4 5 5 3 5 2
```

```
print_clusters <- function(labels, k) {
  for(i in 1:k) {
    print(paste("cluster", i))

print(protein[labels==i,c("Country","RedMeat","Fish","Fr.Veg")])
  }
}
```

```
groups <- pclusters$cluster


print_clusters(groups, 5)
```

```
## [1] "cluster 1"
##     Country RedMeat Fish Fr.Veg
## 10   Greece    10.2  5.9    6.5
## 13    Italy     9.0  3.4    6.7
## 17 Portugal     6.2 14.2    7.9
## 19    Spain     7.1  7.0    7.2
## [1] "cluster 2"
##       Country RedMeat Fish Fr.Veg
## 1     Albania    10.1  0.2    1.7
## 4    Bulgaria     7.8  1.2    4.2
## 18    Romania     6.2  1.0    2.8
## 25 Yugoslavia     4.4  0.6    3.2
## [1] "cluster 3"
##           Country RedMeat Fish Fr.Veg
## 5  Czechoslovakia     9.7  2.0    4.0
## 7       E Germany     8.4  5.4    3.6
## 11        Hungary     5.3  0.3    4.2
## 16         Poland     6.9  3.0    6.6
## 23           USSR     9.3  3.0    2.9
...
```

Good Clustering?

- Total Within Sum of Squares (WSS)
  - The within sum of squares for a single cluster
    - the average squared distance of each point in the cluster from the cluster's centroid.
  - The total within sum of squares
    - the sum of the within sum of squares of all the clusters.

Calinski-Harabasz index

$$\frac{BSS(k)/_{k-1}}{WSS(k)/_{n-k}}$$

- n is the number of data points in the dataset
- k is the number of clusters
- *total sum of squares (TSS)* is the squared distance of all the data points from the dataset's centroid
  - independent of the clustering
- *between sum of squares* BSS(*k*)
  - BSS(k) = TSS - WSS(k)
  - BSS(k) measures how far apart the clusters are from each other
- A good clustering has a small WSS(*k*) and a large BSS(*k*)

WSS and ch-index

```
pclusters$withinss
```

```
## [1] 18.925874  8.012133 16.994661  5.900318 22.110431
```

```
(wss <- sum(pclusters$withinss))
```

```
## [1] 71.94342
```

```
(tss <- pclusters$totss)
```

```
## [1] 216
```

```
(bss <- pclusters$betweenss)
```

```
## [1] 144.0566
```

```
bss == tss - wss
```

```
## [1] TRUE
```

```
ch.index <- (bss/(k-1)) / (wss/(ndata-k))
print(sprintf('CH index of this clustering is %.f', ch.index))
```

```
## [1] "CH index of this clustering is 57"
```

Practice

- Calculate CH index and WSS with different k = 1, 2, ..., 10 for k-means clustering and draw line plot

Hierarchical Clustering

data loading and preparation

```
protein <- read.table("protein.txt", sep="\t", header=TRUE)
summary(protein)
##           Country      RedMeat        WhiteMeat          Eggs
##  Albania       : 1   Min.   : 4.400   Min.   : 1.400   Min.   :0.500
##  Austria       : 1   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.700
##  Belgium       : 1   Median : 9.500   Median : 7.800   Median :2.900
##  Bulgaria      : 1   Mean   : 9.828   Mean   : 7.896   Mean   :2.936
##  Czechoslovakia: 1   3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.700
##  Denmark       : 1   Max.   :18.000   Max.   :14.000   Max.   :4.700
##  (Other)       :19
##       Milk            Fish           Cereals          Starch
##  Min.   : 4.90   Min.   : 0.200   Min.   :18.60   Min.   :0.600
##  1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30   1st Qu.:3.100
##  Median :17.60   Median : 3.400   Median :28.00   Median :4.700
##  Mean   :17.11   Mean   : 4.284   Mean   :32.25   Mean   :4.276
##  3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10   3rd Qu.:5.700
##  Max.   :33.70   Max.   :14.200   Max.   :56.70   Max.   :6.500
##
##       Nuts           Fr.Veg
##  Min.   :0.700   Min.   :1.400
##  1st Qu.:1.500   1st Qu.:2.900
##  Median :2.400   Median :3.800
```

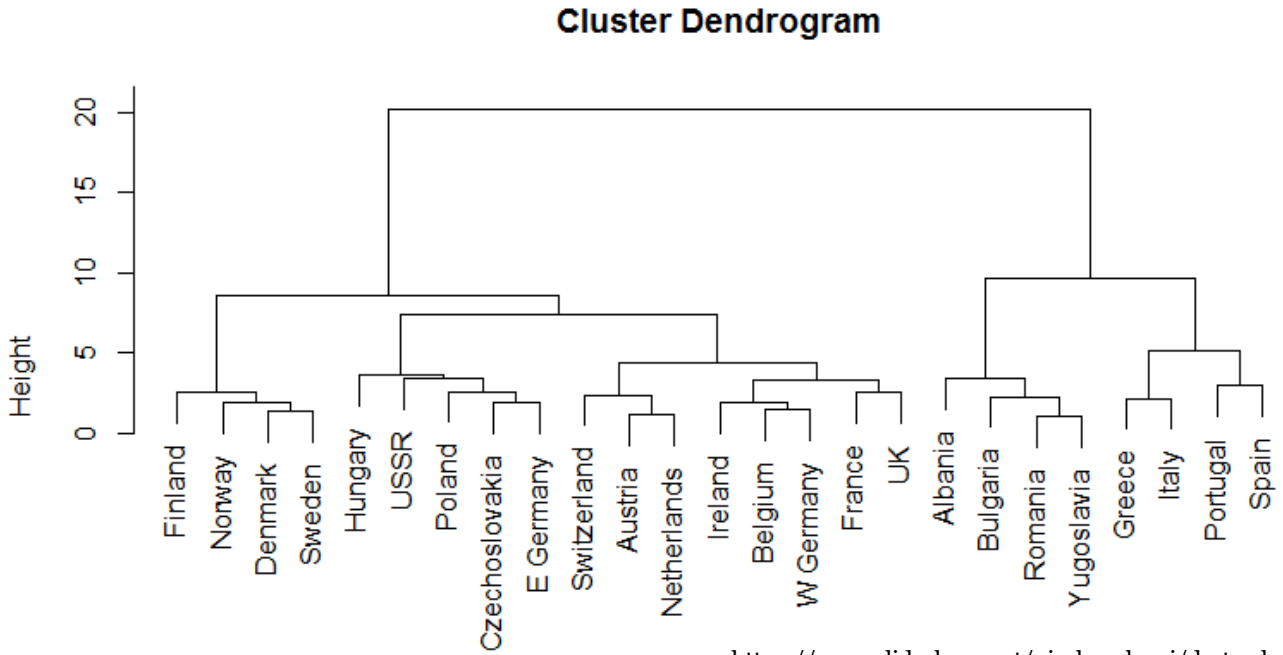
```
vars.to.use <- colnames(protein)[-1]
pmatrix <- scale(protein[,vars.to.use])
pcenter <- attr(pmatrix, "scaled:center")
pscale <- attr(pmatrix, "scaled:scale")
```

protein dataset from 1973 on protein consumption from nine different food groups in 25 countries in Europe.

hierachical clustering

```
d <- dist(pmatrix, method="euclidean")
pfit <- hclust(d, method="ward.D")
plot(pfit, labels=protein$Country)
```

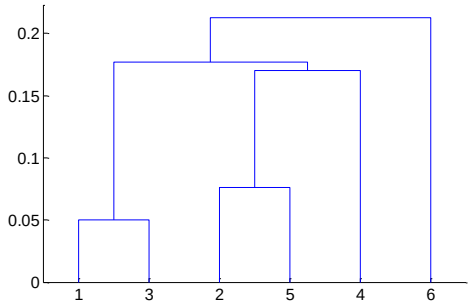
ward:
For each data point as an individual cluster,
merges clusters iteratively so as to minimize the *total within sum of squares (WSS)* of the clustering
http://rfriend.tistory.com/227

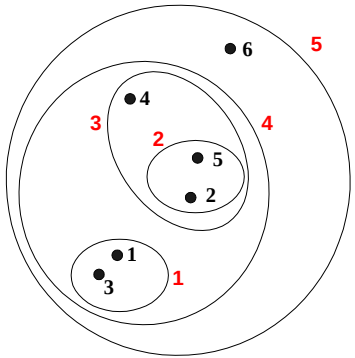


https://www.slideshare.net/pierluca.lanzi/dmtm-lecture-12-hierarchical-clustering
https://onlinecourses.science.psu.edu/stat505/node/146/

Hierarchical Clustering

- Produces a set of ***nested clusters*** organized as a hierarchical tree
- Can be visualized as a **dendrogram**
  - A tree-like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

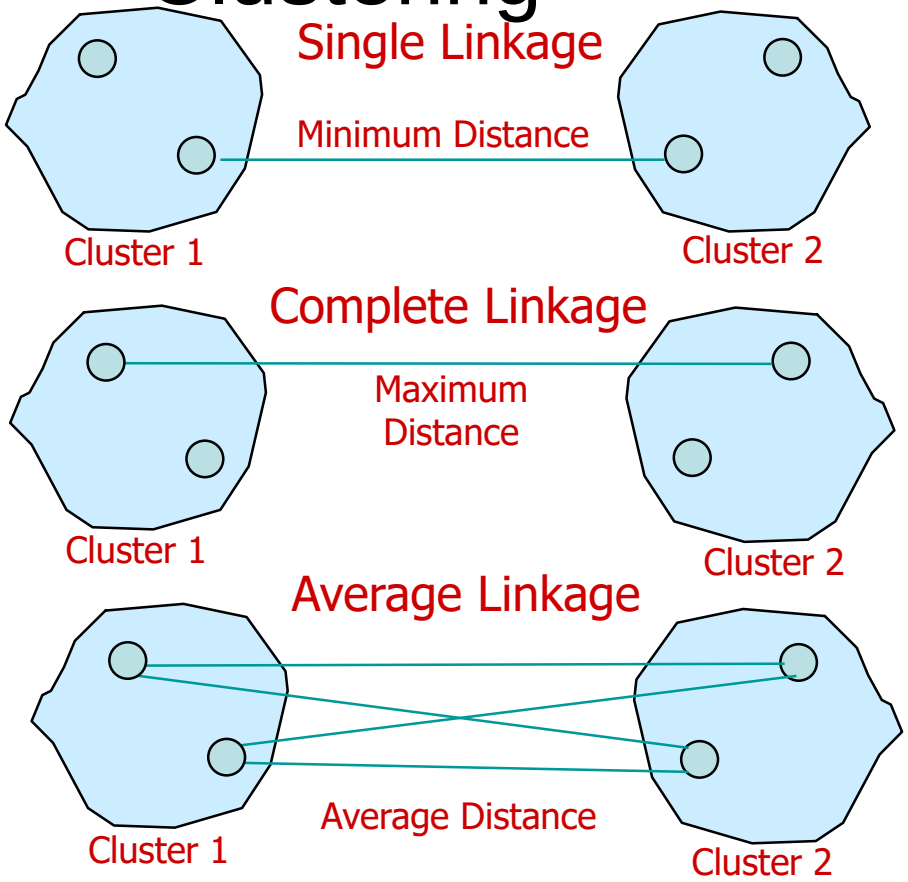
- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

Hierarchical Agglomerative Clustering- Linkage Method

- The **single linkage** method is based on minimum distance, or the nearest neighbor rule.
- The **complete linkage** method is based on the maximum distance or the furthest neighbor approach.
- The **average linkage** method the distance between two clusters is defined as the average of the distances between all pairs of objects

Linkage Methods of Clustering

Fig. 20.5



Hierarchical Agglomerative Clustering- Variance and Centroid Method

- **Variance methods** generate clusters to minimize the within-cluster variance.

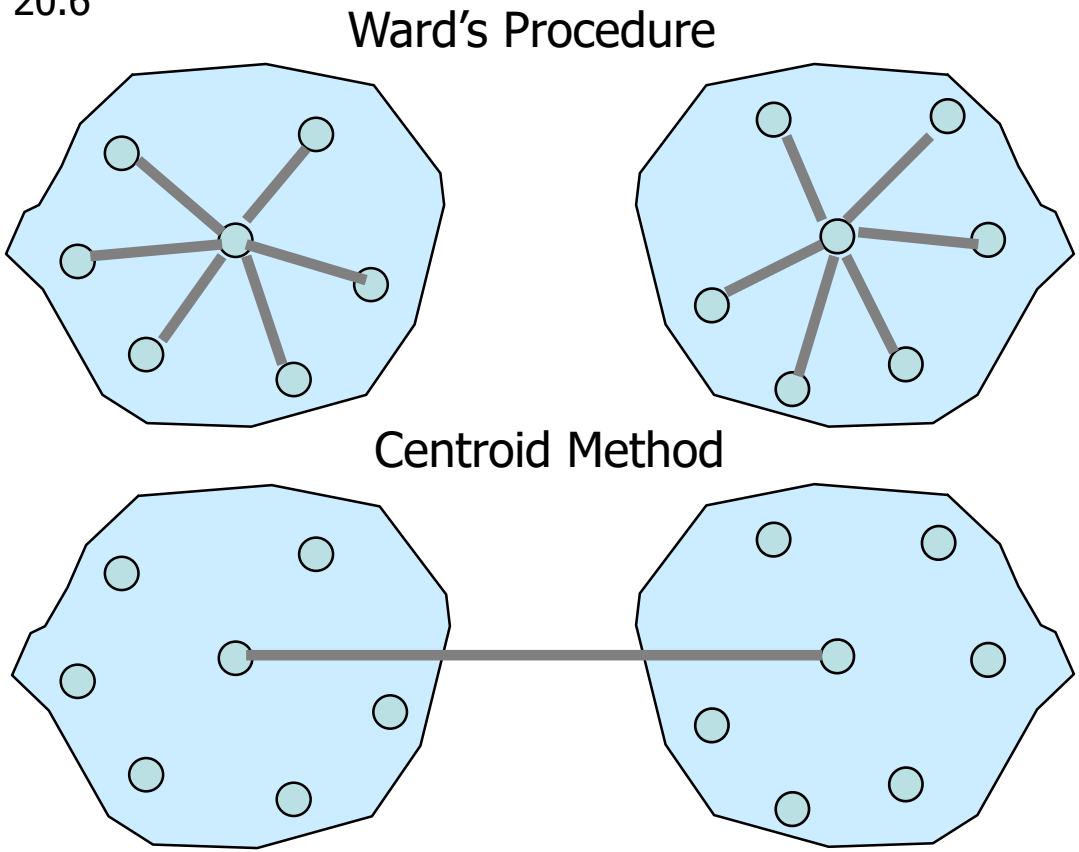
- **Ward's procedure** is commonly used. For each cluster, the sum of squares is calculated. The two clusters with the smallest increase in the overall sum of squares within cluster distances are combined.

- In the **centroid methods**, the distance between two clusters is the distance between their centroids (means for all the variables),

- Of the hierarchical methods, average linkage and Ward's methods have been shown to perform better than the other procedures.

Other Agglomerative Clustering Methods

Fig. 20.6

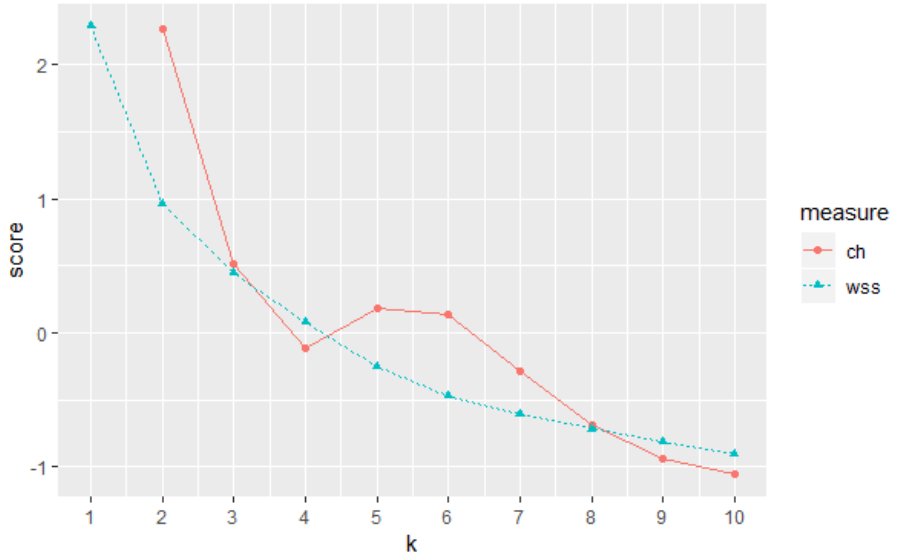


```r
groups <- cutree(pfit, k=5)

print_clusters <- function(labels, k) {                    # Note: 1
  for(i in 1:k) {
    print(paste("cluster", i))
    print(protein[labels==i,c("Country","RedMeat","Fish","Fr.Veg")])
  }
}

print_clusters(groups, 5)
```

```
## [1] "cluster 1"
##       Country RedMeat Fish Fr.Veg
## 1     Albania    10.1  0.2    1.7
## 4    Bulgaria     7.8  1.2    4.2
## 18    Romania     6.2  1.0    2.8
## 25 Yugoslavia     4.4  0.6    3.2
## [1] "cluster 2"
##        Country RedMeat Fish Fr.Veg
## 2      Austria     8.9  2.1    4.3
## 3      Belgium    13.5  4.5    4.0
## 9       France    18.0  5.7    6.5
## 12     Ireland    13.9  2.2    2.9
## 14 Netherlands     9.5  2.5    3.7
## 21 Switzerland    13.1  2.3    4.9
## 22          UK    17.4  4.3    3.3
## 24   W Germany    11.4  3.4    3.8
## [1] "cluster 3"
##           Country RedMeat Fish Fr.Veg
## 5  Czechoslovakia     9.7  2.0    4.0
## 7        E Germany     8.4  5.4    3.6
## 11         Hungary     5.3  0.3    4.2
## 16          Poland     6.9  3.0    6.6
## 23            USSR     9.3  3.0    2.9
## [1] "cluster 4"
##    Country RedMeat Fish Fr.Veg
## 6  Denmark    10.6  9.9    2.4
## 8  Finland     9.5  5.8    1.4
## 15  Norway     9.4  9.7    2.7
## 20  Sweden     9.9  7.5    2.0
## [1] "cluster 5"
##     Country RedMeat Fish Fr.Veg
## 10   Greece    10.2  5.9    6.5
## 13    Italy     9.0  3.4    6.7
## 17 Portugal     6.2 14.2    7.9
## 19    Spain     7.1  7.0    7.2
```

Practice

```
##      k          ch          wss
## 1    1         NaN  2.28417365
## 2    2  2.2615170  0.96519707
## 3    3  0.5112068  0.43992959
## 4    4 -0.1195726  0.07640784
## 5    5  0.1831375 -0.25182002
## 6    6  0.1389624 -0.46859363
## 7    7 -0.2930582 -0.60623273
## 8    8 -0.6844449 -0.71674946
## 9    9 -0.9388138 -0.81561458
## 10 10 -1.0589343 -0.90669774
```



- Calculate CH index and WSS with different k = 1, 2, ..., 10 for hierarchical clustering

Homework

- Survey various types of Distance Measure and when to use them properly
- Ref: https://www.datanovia.com/en/lessons/clustering-distance-measures/

Reference

- MacQueen, J. (1967, June). **Some methods for classification and analysis of multivariate observations**. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).
- Madhu Sanjeevi (blog)**, Chapter 4: Decision Trees Algorithms,** https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1
- David Sontag, New York University, **Decision trees Lecture 11,** https://people.csail.mit.edu/dsontag/courses/ml16/slides/lecture11.pdf
- Breiman, L. (1996). **Bagging predictors.** Machine learning, 24(2), 123-140.
- Breiman, L. (2001). **Random forests.** Machine learning, 45(1), 5-32.
- Liaw, A., & Wiener, M. (2002). **Classification and regression by randomForest**. R news, 2(3), 18-22.
- Scott Fortmann-Roe, **Understanding the Bias-Variance Tradeoff**, http://scott.fortmann-roe.com/docs/BiasVariance.html