浙江大学爱丁堡大学联合学院

ZJU-UoE Institute

Unsupervised learning methods

Nicola Romanò - nicola.romano@ed.ac.uk

Learning objectives

At the end of this lecture you should be able to:

- Explain the difference between supervised and unsupervised learning
- Give examples of when these methods can be used
- Explain the k-means and hierarchical clustering methods and discuss their advantages and disadvantages

Machine learning in biology

Some example problems we would like to be able to address

Machine learning in biology

Some example problems we would like to be able to address

1. We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.

Machine learning in biology

Some example problems we would like to be able to address

1. We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.
2. We measure hormonal levels in healthy controls or patients with an illness. Given measures from a new individual, can we predict whether they are healthy or ill?

Machine learning in biology

Some example problems we would like to be able to address

1. We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.
2. We measure hormonal levels in healthy controls or patients with an illness. Given measures from a new individual, can we predict whether they are healthy or ill?
3. Given a set of photos of cells we want to divide depending on their shape.

Some example problems we would like to be able to address

1. We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.
2. We measure hormonal levels in healthy controls or patients with an illness. Given measures from a new individual, can we predict whether they are healthy or ill?
3. Given a set of photos of cells we want to divide depending on their shape.
4. Given measurements of expression of thousands of different genes from some tumour samples, we want to know whether there are specific classes of tumours, defined by a precise genetic signature.

Machine learning in biology

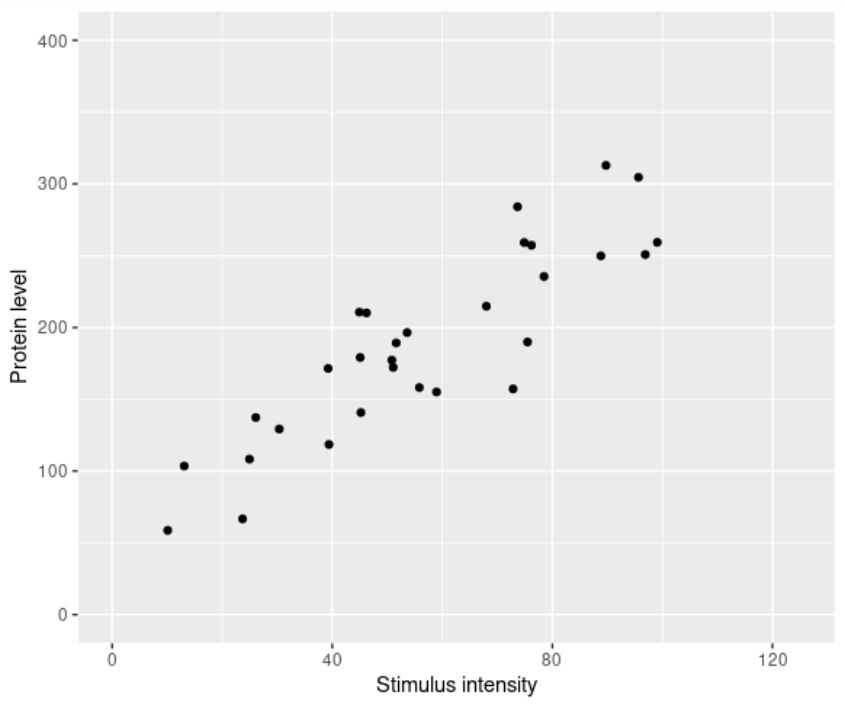
Some example problems we would like to be able to address

1. We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.
2. We measure hormonal levels in healthy controls or patients with an illness. Given measures from a new individual, can we predict whether they are healthy or ill?
3. Given a set of photos of cells we want to divide depending on their shape.
4. Given measurements of expression of thousands of different genes from some tumour samples, we want to know whether there are specific classes of tumours, defined by a precise genetic signature.

How do we solve these problems?

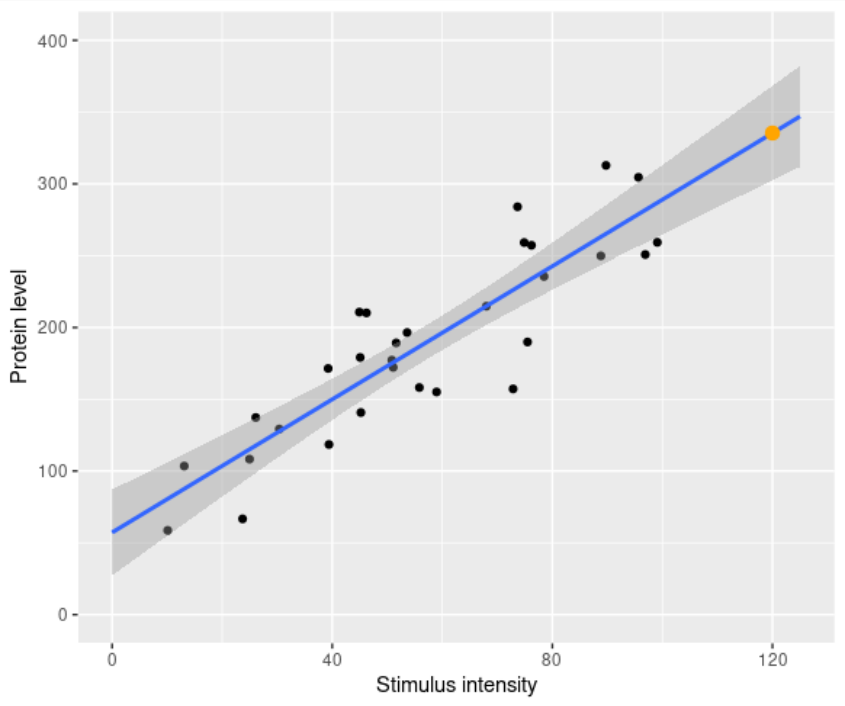
Problem 1

We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.



Problem 1

We measure the amount of protein produced by a cell line in response to different intensities of a stimulus. We would like to predict the amount of protein after a stimulus of a different intensity.

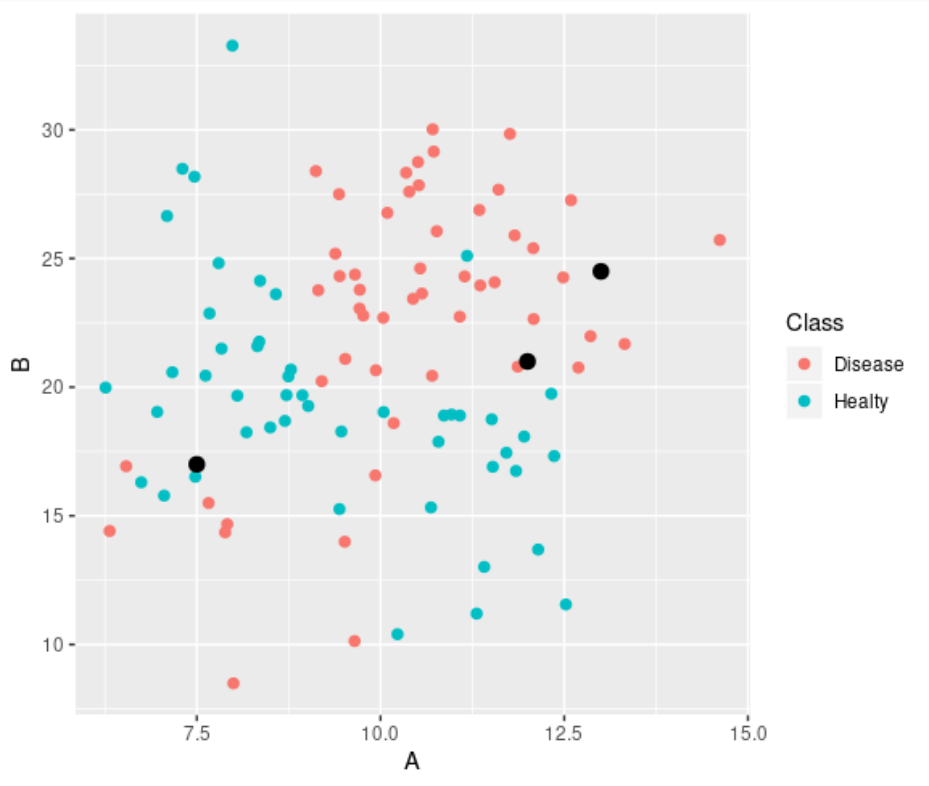


Regression

```
lm(protein ~ stimulus)
```

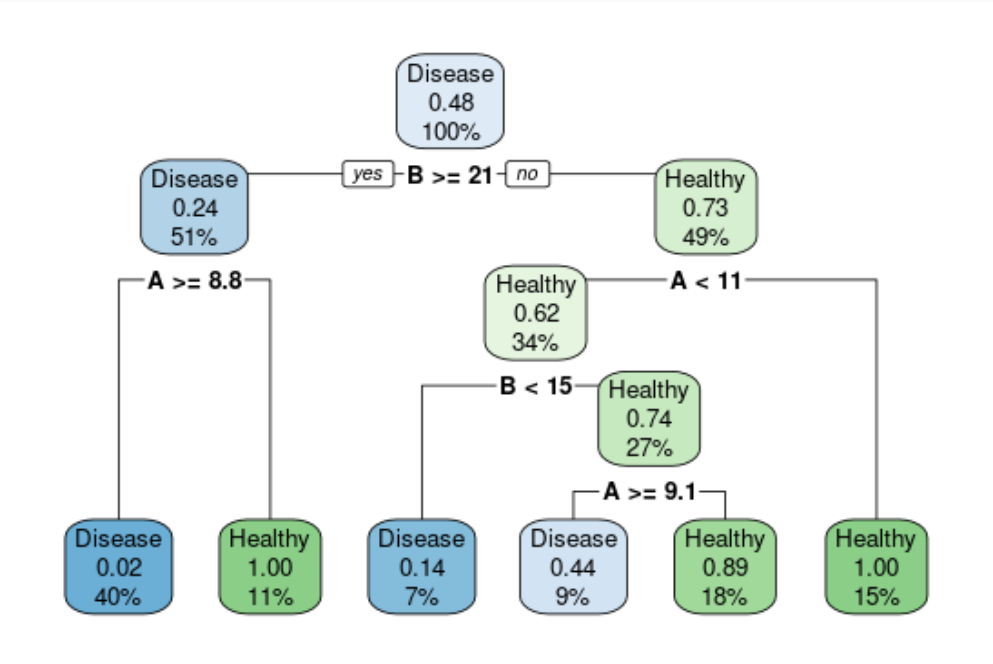
Problem 2

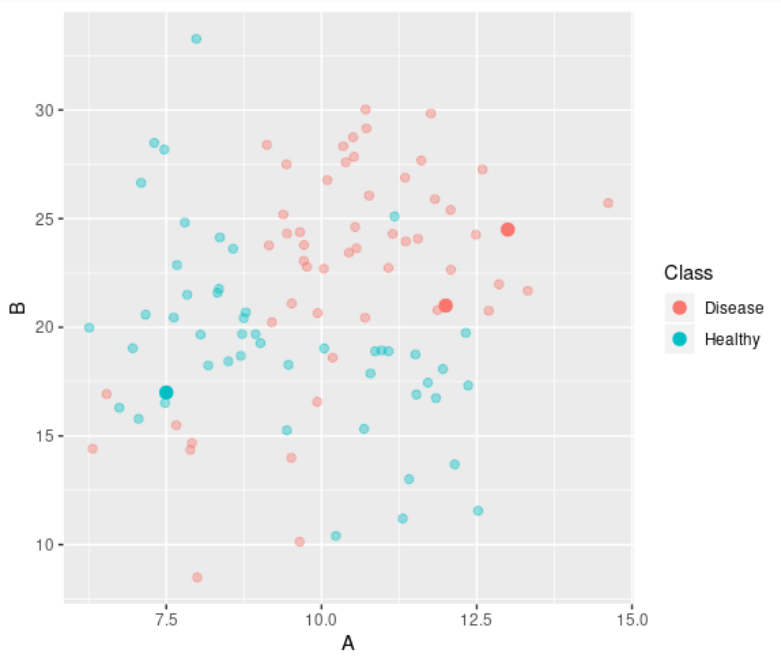
We measure hormonal levels in healthy controls or patients with an illness. Given measures from a new individual, can we predict whether they are healthy or ill?



Problem 2

We measure hormonal levels in healthy controls or patients with an illness. Given measures from a new individual, can we predict whether they are healthy or ill?



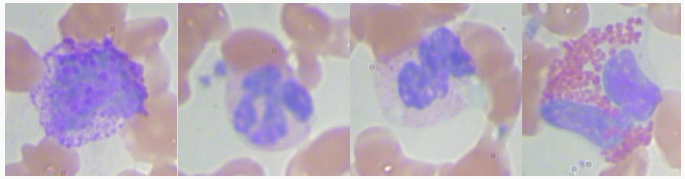


Classification

```
rpart(Class ~ A + B)
```

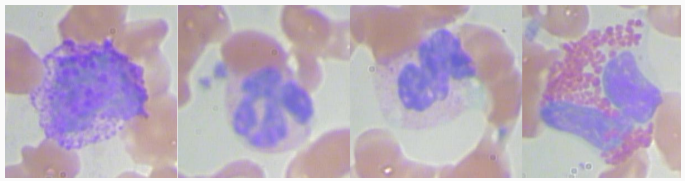
Problem 3 and 4

- Given a set of photos of cells we want to divide depending on their shape.

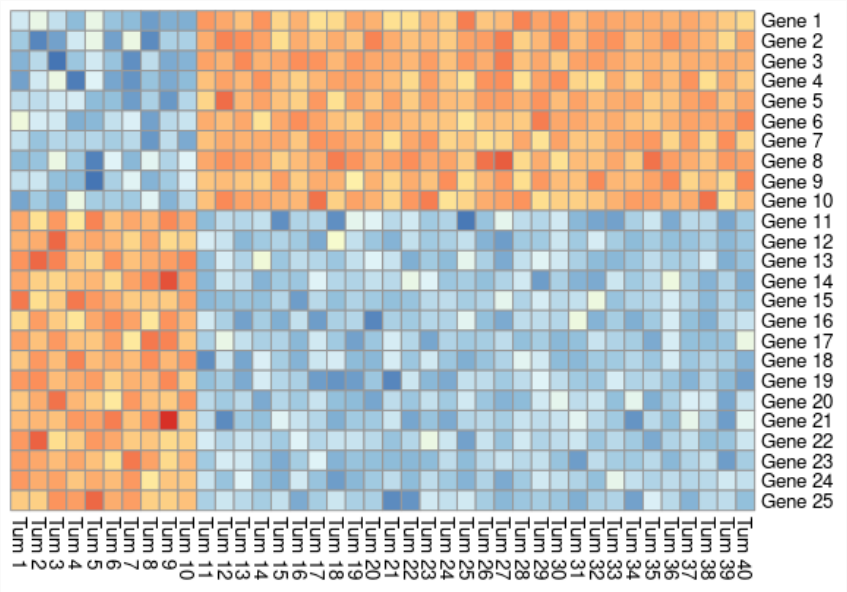


Problem 3 and 4

- Given a set of photos of cells we want to divide depending on their shape.

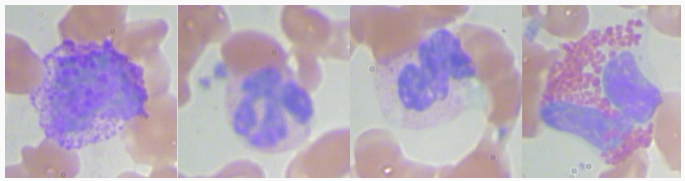


  - Given measurements of expression of thousands of different genes from some tumour samples, we want to know whether there are specific classes of tumours, defined by a precise genetic signature.

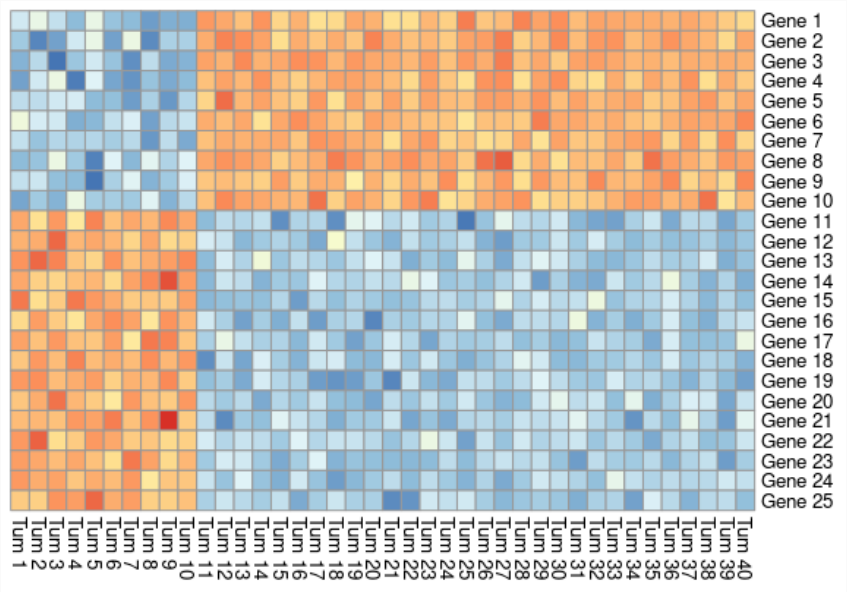


Problem 3 and 4

- Given a set of photos of cells we want to divide depending on their shape.



  - Given measurements of expression of thousands of different genes from some tumour samples, we want to know whether there are specific classes of tumours, defined by a precise genetic signature.



How to do this? We don't have information to train a model.

Supervised vs unsupervised learning

Machine learning methods can be broadly divided into supervised and unsupervised.

Supervised methods

- We train a model using a training set with known labels
- We test the accuracy of the model on a test set with known labels, but that we did not use for training.
- We can use the model for prediction/classification.
- Examples: binary trees, RandomForest, SVM, Neural Networks, ...

Supervised vs unsupervised learning

Machine learning methods can be broadly divided into supervised and unsupervised.

Supervised methods

- We train a model using a training set with known labels
- We test the accuracy of the model on a test set with known labels, but that we did not use for training.
- We can use the model for prediction/classification.
- Examples: binary trees, RandomForest, SVM, Neural Networks, ...

Unsupervised methods

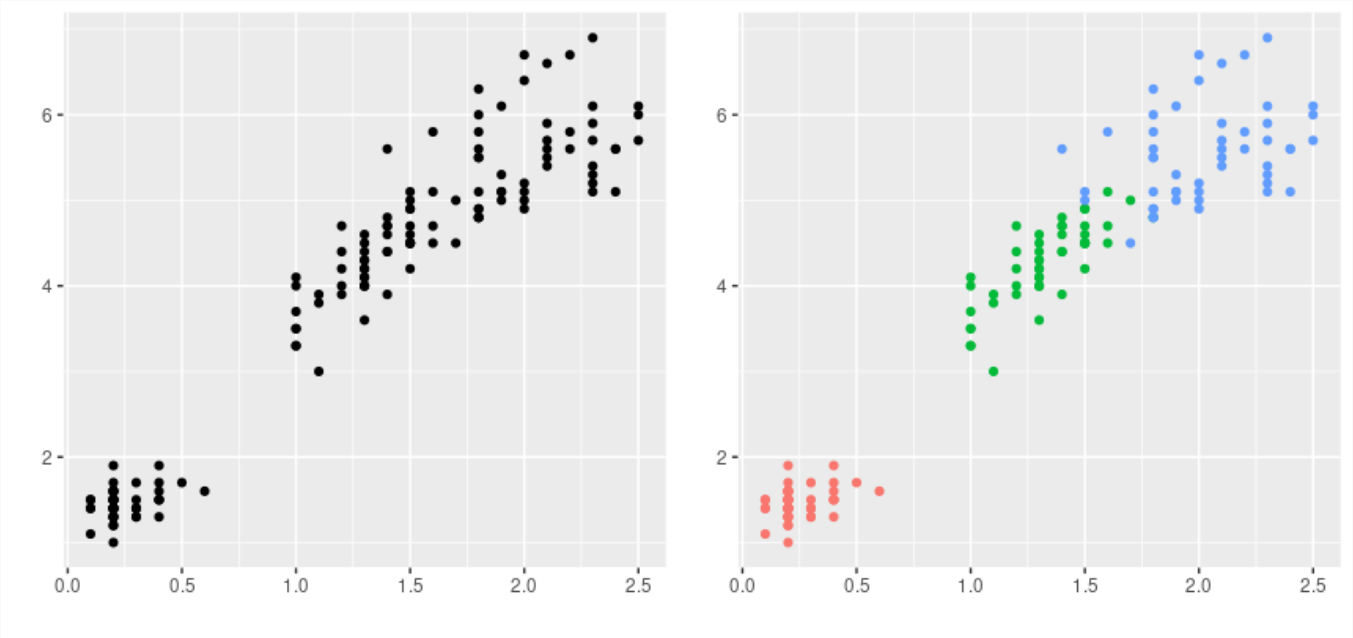
- We have an unlabelled dataset.
- We use a model to find data patterns/groupings (**clustering**).
- Examples: k-means, hierarchical clustering (this lecture), dimensionality reduction (next lecture)...

Clustering

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). (*Wikipedia*)

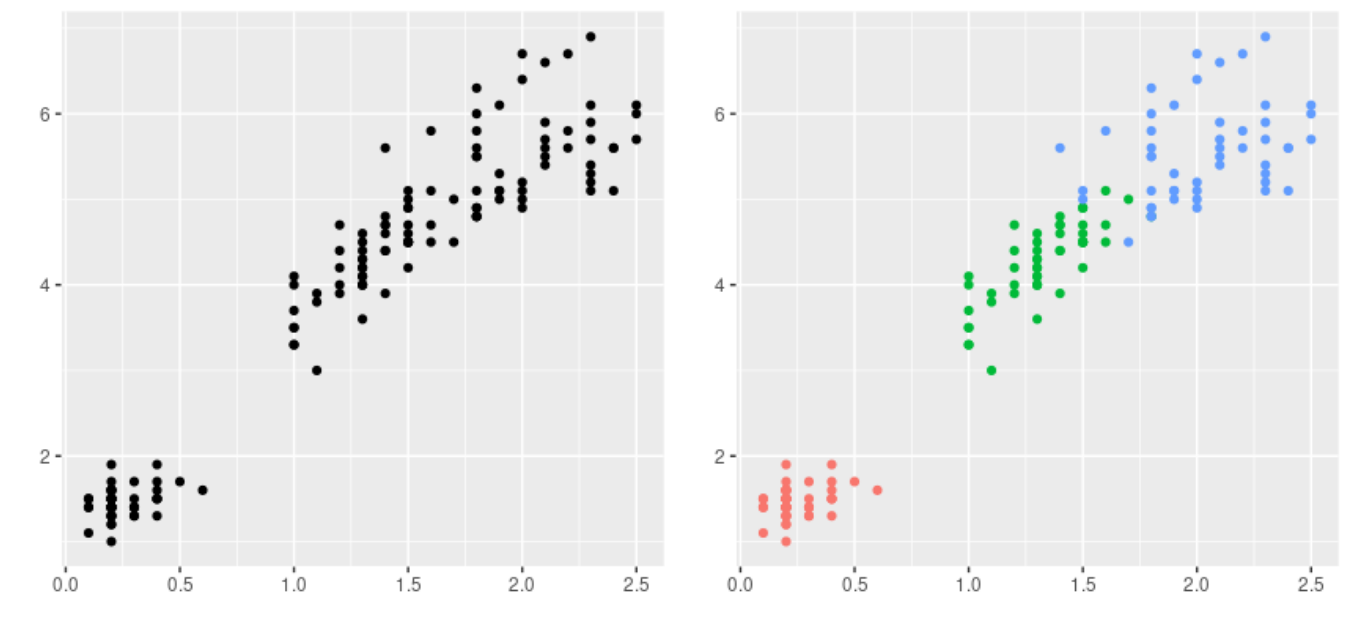
Clustering

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). (*Wikipedia*)



Clustering

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). (*Wikipedia*)



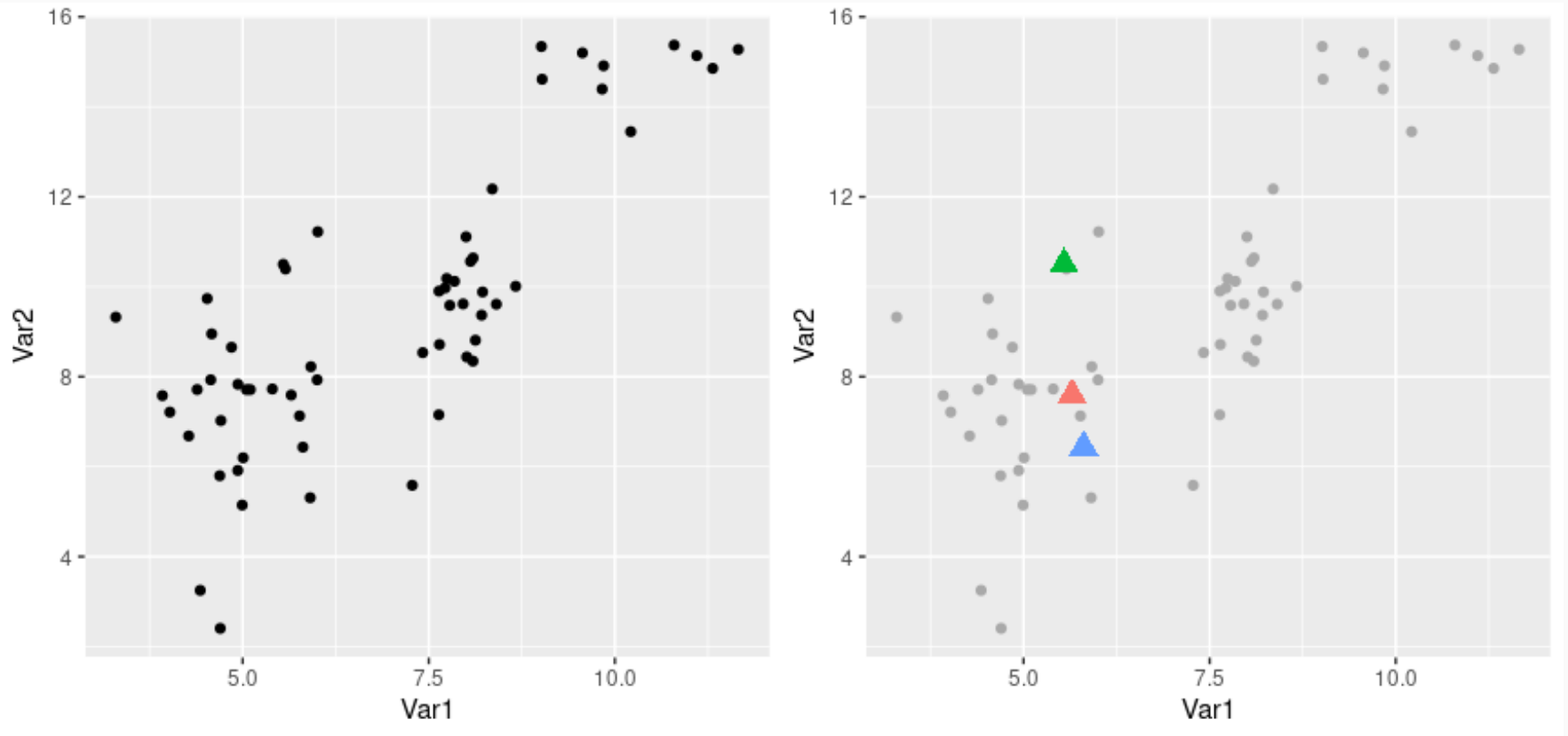
The number of clusters in an unknown data set is not trivial to determine.

The k-means algorithm

- One of the simplest approaches to clustering
- It's an iterative algorithm that divides the dataset in k clusters

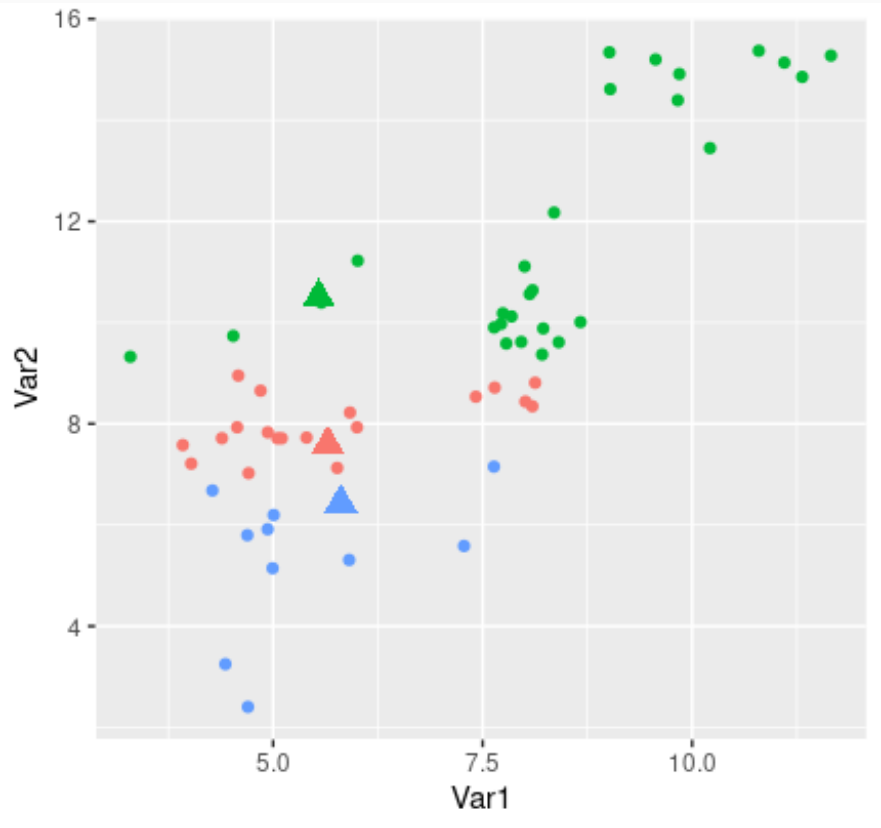
k-means - Step 1

- We select k random points as starting centers (called *centroids*)
- We assign each point in the dataset to the closest centroid, thus defining k clusters



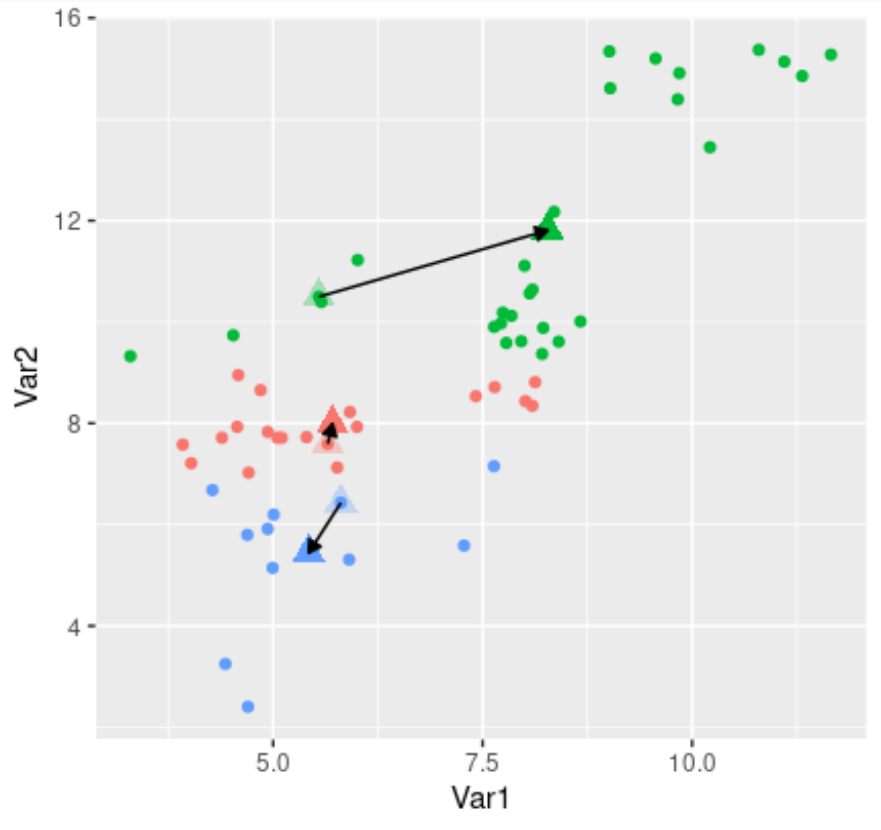
k-means - Step 1

- We select k random points as starting centers (called *centroids*)
- We assign each point in the dataset to the closest centroid, thus defining k clusters



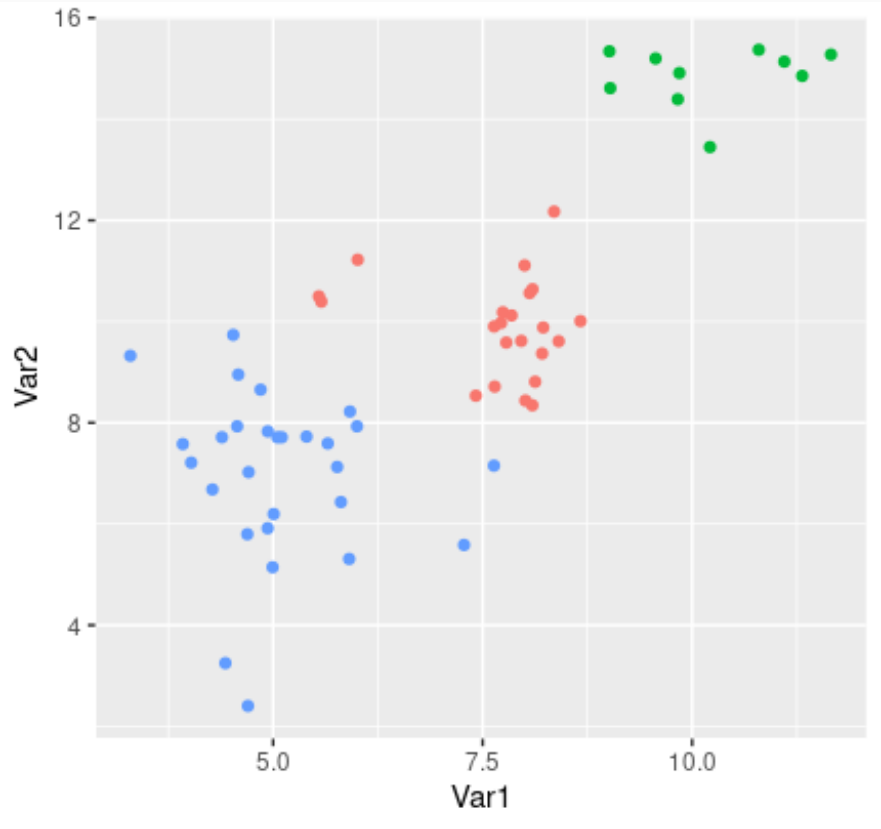
k-means - Step 2

- We move the centroids to the center of each cluster
- We reassign cluster memberships and continue repeating until clusters don't change anymore or until we reach a certain number of iterations
- Most often k-means converges after 10-20 iterations



k-means final clustering

Our final result



How to do k-means in R?

It's really simple!

```
km <- kmeans(mydata, centers=3)
# We can choose multiple sets of starting centroids
km <- kmeans(mydata, centers=3, nstart=50)
# Clusters can be found in km$cluster
```

We will see some use of this in Workshop 7!

k-means pros and cons

Advantages

- Generally fast
- Computationally easy to implement

Disadvantages

- Results are heavily dependent on the random choice of centroids at the start
- Need to specify the number of clusters in advance
- Works better with equally sized clusters
- Sensible to outliers

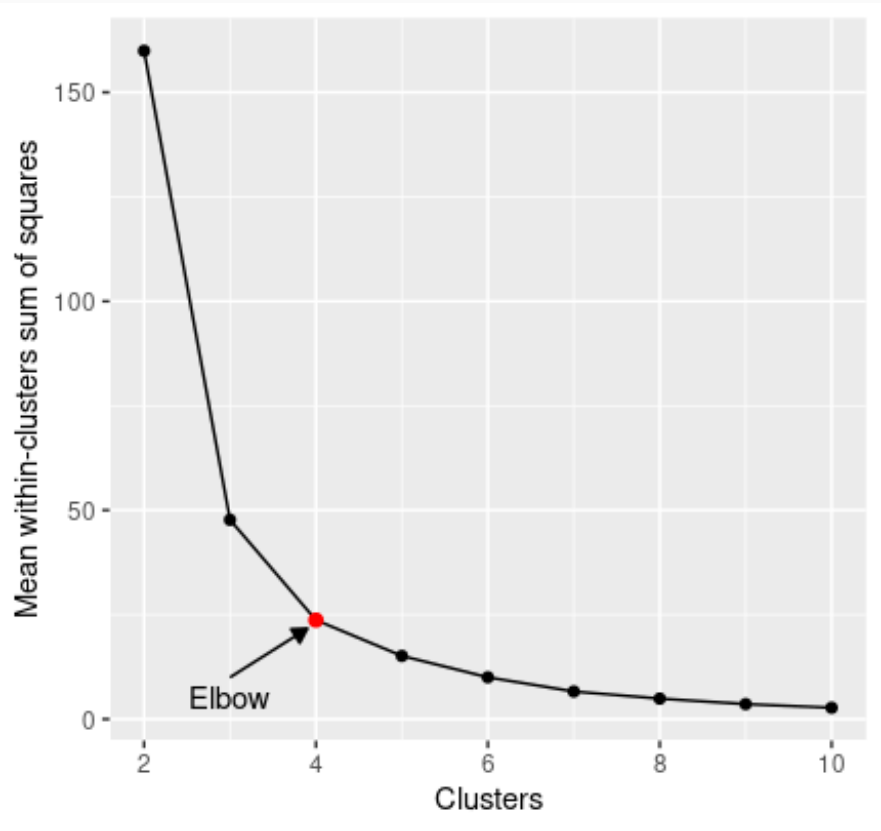
Determining number of clusters

- Determining the number of clusters is difficult.
- Depends on the question you are asking
- There is no *correct* solution

Determining number of clusters

- Determining the number of clusters is difficult.
- Depends on the question you are asking
- There is no *correct* solution
- Empirical method - **elbow plot**

```
km$tot.withinss / n.cluster
```



Alternative strategy - hierarchical clustering

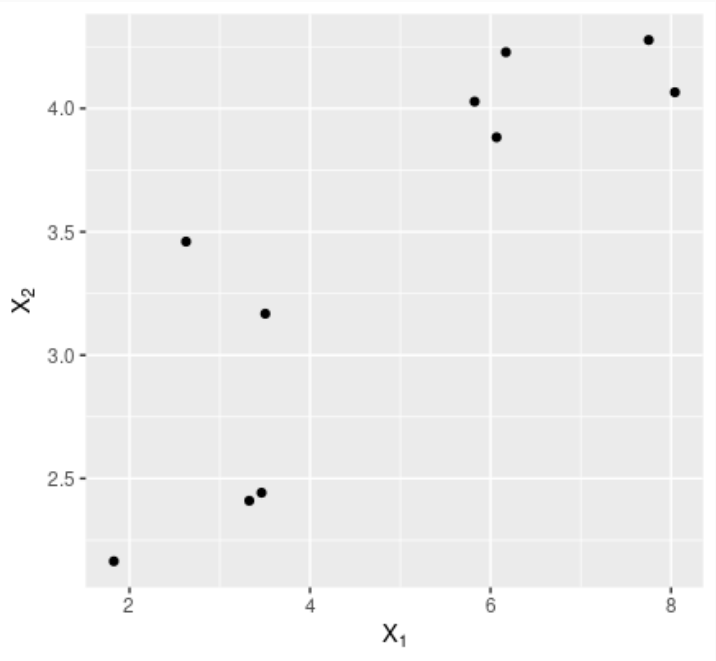
Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. (*Wikipedia*)

Two main strategies:

- Agglomerative or "bottom-up" hierarchical clustering initially creates one cluster per observation and then merge them depending on their similarity
- Divisive or "top-down" hierarchical clustering puts all observations in one cluster then recursively splits the cluster.
- Generates a *dendrogram* (tree like plot)

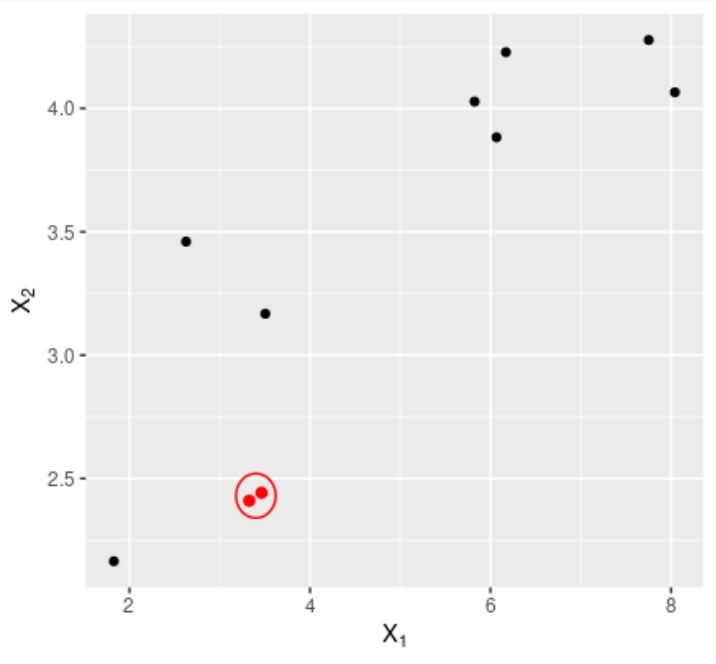
Agglomerative hierarchical clustering

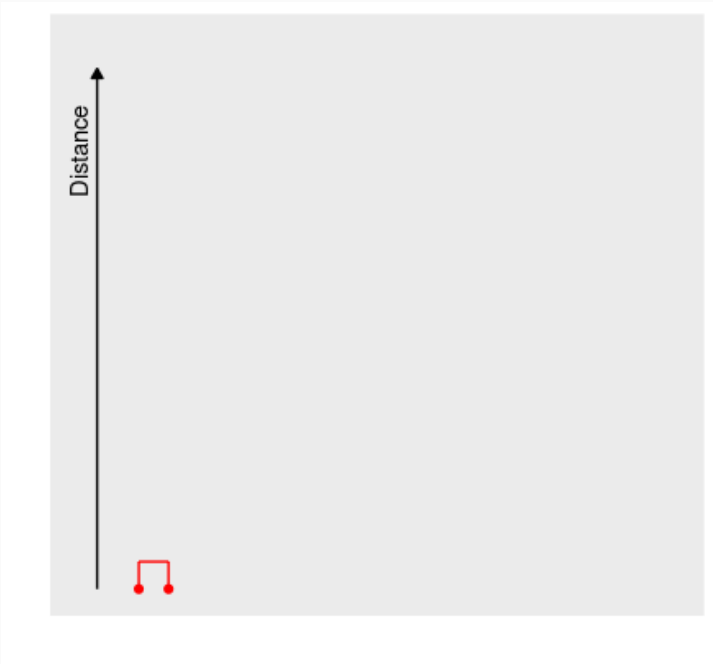
- We start from n data points, each in a clusters
- We define some distance metrics (see later)
- We find the pair of points with the smaller distance
- We start building our dendrogram



Agglomerative hierarchical clustering

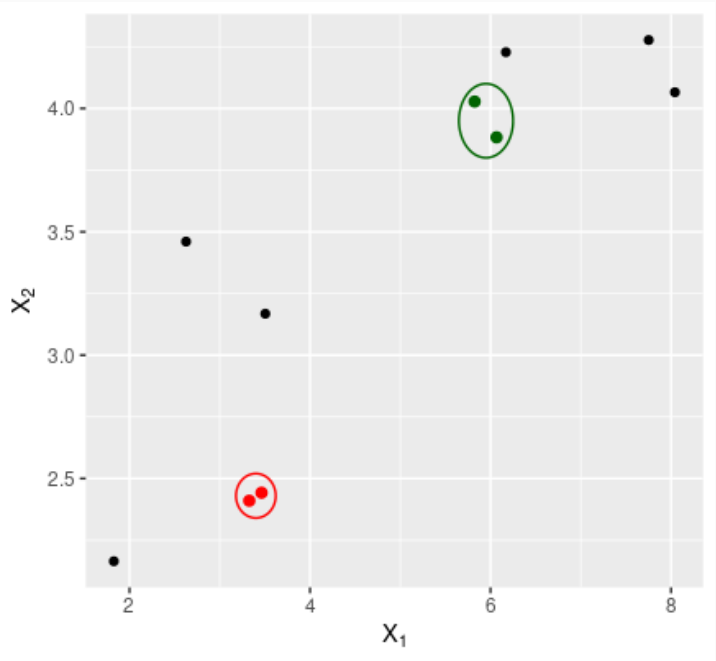
- We start from n data points, each in a clusters
- We define some distance metrics (see later)
- We find the pair of points with the smaller distance
- We start building our dendrogram

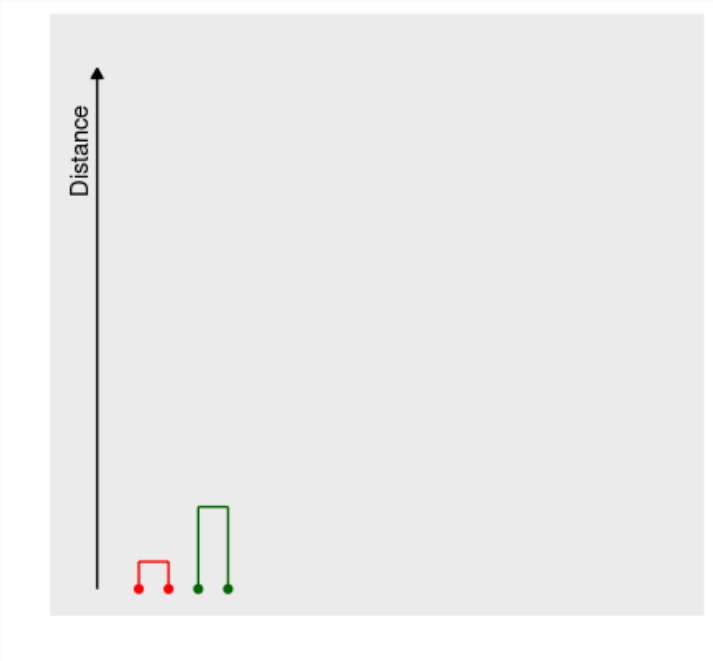




Agglomerative hierarchical clustering

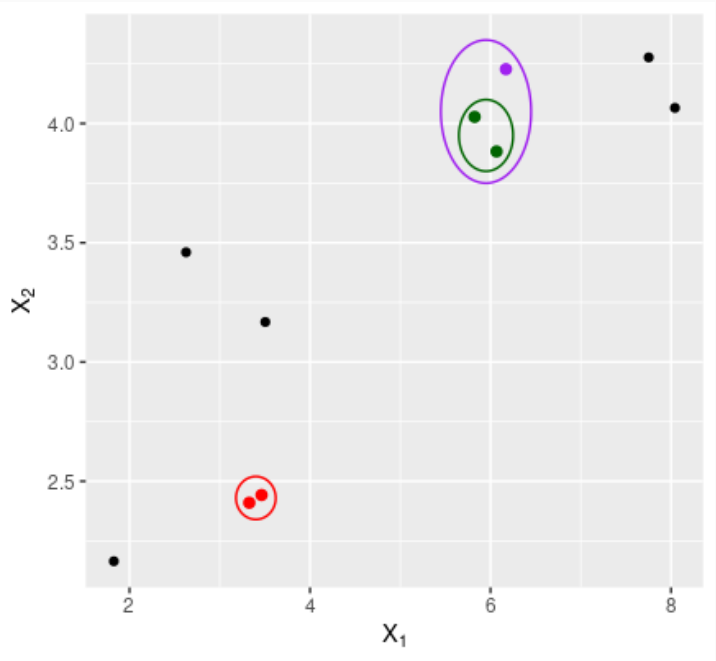
- We start from n data points, each in a clusters
- We define some distance metrics (see later)
- We find the pair of points with the smaller distance
- We start building our dendrogram

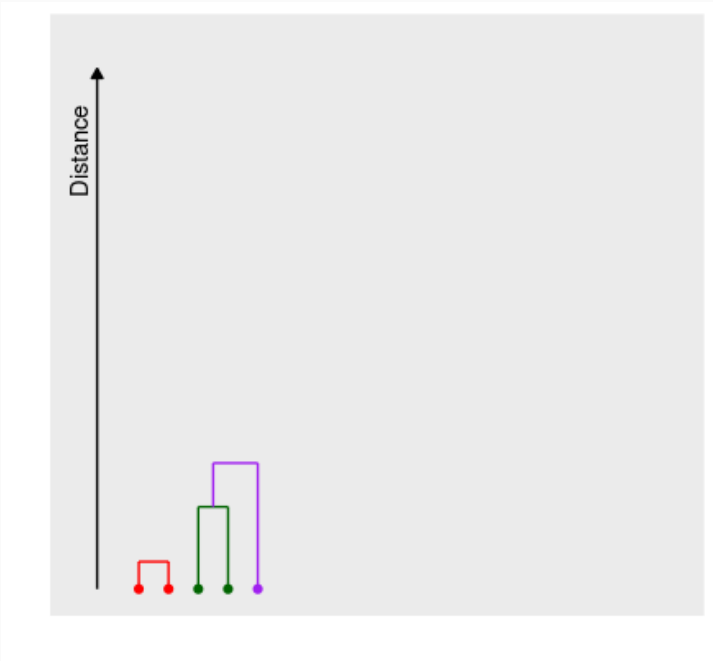




Agglomerative hierarchical clustering

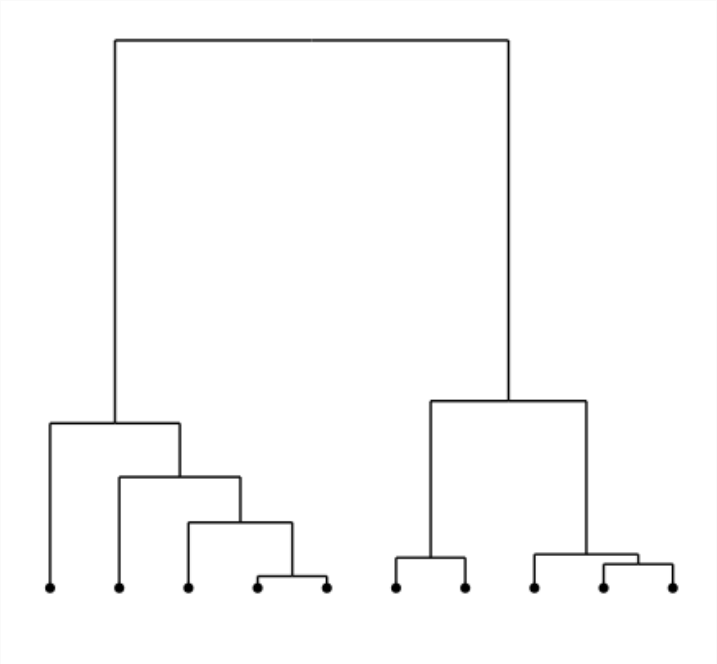
- We start from n data points, each in a clusters
- We define some distance metrics (see later)
- We find the pair of points with the smaller distance
- We start building our dendrogram





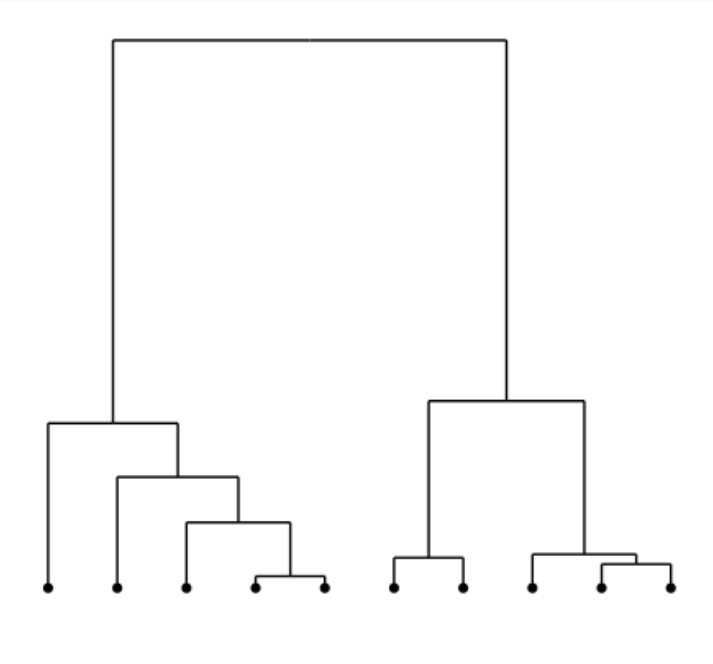
Agglomerative hierarchical clustering

... and so on until every point has been clustered!



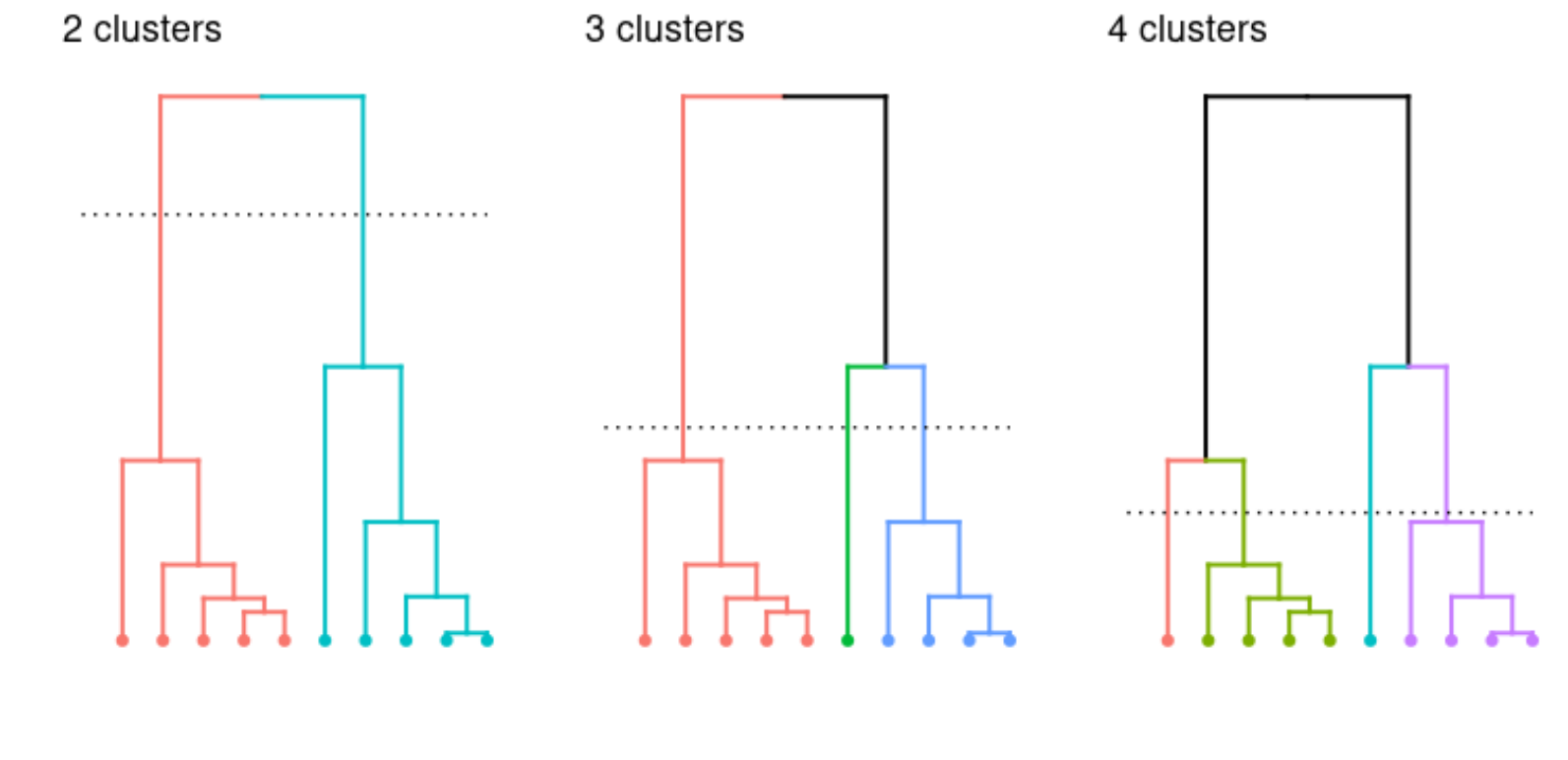
Agglomerative hierarchical clustering

... and so on until every point has been clustered!



```
pt.dist <- dist(hc, method = "euclidean")
hc <- hclust(pt.dist, method = "complete")
```

How many clusters do we have?



Distance methods

```
pt.dist <- dist(hc, method = "euclidean")
```

- The `method` parameter defines the metrics used for calculating the distance between data points
- Several available (`"euclidean"`, `"maximum"`, `"manhattan"`, `...`)

Distance methods

```
pt.dist <- dist(hc, method = "euclidean")
```

- The `method` parameter defines the metrics used for calculating the distance between data points
- Several available (`"euclidean"`, `"maximum"`, `"manhattan"`, ...)
- Given two data points A and B: $A(X1_A, X2_A, ..., Xn_A), B(X1_B, X2_B, ..., Xn_B)$
  Euclidean distance
  $d = \sqrt{(X1_A - X1_B)^2 + (X2_A - X2_B)^2 + ... + (Xn_A - Xn_B)^2}$

Distance methods

```
pt.dist <- dist(hc, method = "euclidean")
```

- The `method` parameter defines the metrics used for calculating the distance between data points
- Several available (`"euclidean"`, `"maximum"`, `"manhattan"`, ...)
- Given two data points A and B: $A(X1_A, X2_A, ..., Xn_A), B(X1_B, X2_B, ..., Xn_B)$
  Euclidean distance
  $d = \sqrt{(X1_A - X1_B)^2 + (X2_A - X2_B)^2 + ... + (Xn_A - Xn_B)^2}$
- Maximum distance $d = max\{|X1_A - X1_B|, |X2_A - X2_B|, ..., |Xn_A - Xn_B|\}$

Distance methods

```
pt.dist <- dist(hc, method = "euclidean")
```

- The `method` parameter defines the metrics used for calculating the distance between data points
- Several available (`"euclidean"`, `"maximum"`, `"manhattan"`, ...)
- Given two data points A and B: $A(X1_A, X2_A, ..., Xn_A), B(X1_B, X2_B, ..., Xn_B)$
  Euclidean distance
  $d = \sqrt{(X1_A - X1_B)^2 + (X2_A - X2_B)^2 + ... + (Xn_A - Xn_B)^2}$
- Maximum distance $d = max\{|X1_A - X1_B|, |X2_A - X2_B|, ..., |Xn_A - Xn_B|\}$
- See `?dist` for full list and details

Linkages

```
hc <- hclust(pt.dist, method = "complete")
```

- Similarly, the `method` parameter defines how the dendrogram is built
- Common values are: `complete`, `average`, `ward.D2`
- See `?hclust` for details

Summary

- Statistical models and machine learning algorithms allow us to answer many biological questions
- Choosing the right method to answer the right question is not easy
- Clustering methods are becoming very important, especially when dealing with large dataset and/or data with high dimensionality (more on that in the next two data analysis lectures)
- Many other approaches apart from those we saw today!