Week 11: Clustering and the Dendrogram

Visual Data Analytics

University of Sydney

Outline

- Distance
- Single Linkage
- Other Hierarchical Clustering
- Dendrogram

Motivation

- We can profile an individual according to their attributes.
  - Are two individuals similar?
  - Can we group to individuals together?
  - How can we visualise this?
- The method is hierarchical clustering and the visualisation is the Dendrogram.
- The ideas we cover are useful in marketing and other business problems.

Distance

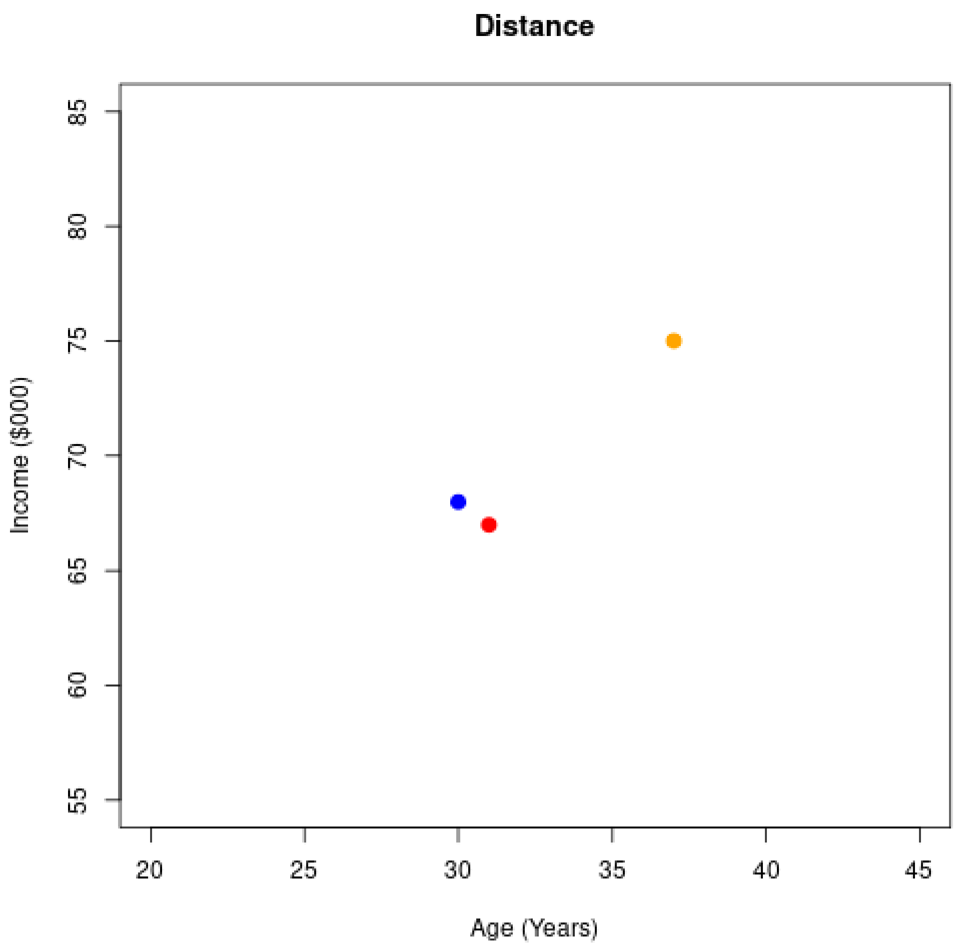
Why distance?

- Many problems that involve thinking about how *similar* or dissimilar two observations are. For example:
  - May use the same marketing strategy for *similar* demographic groups.
  - May lend money to applicants who are *similar* to those who pay debts back.
- Arguably the most important concept in data analysis is *distance*

Simple example

- Consider 3 individuals:
  - Mr Orange: 37 years of age earns $75k a year
  - Mr Red: 31 years of age earns $67k a year
  - Mr Blue: 30 years of age earns $68k a year
- Which two are the most similar?

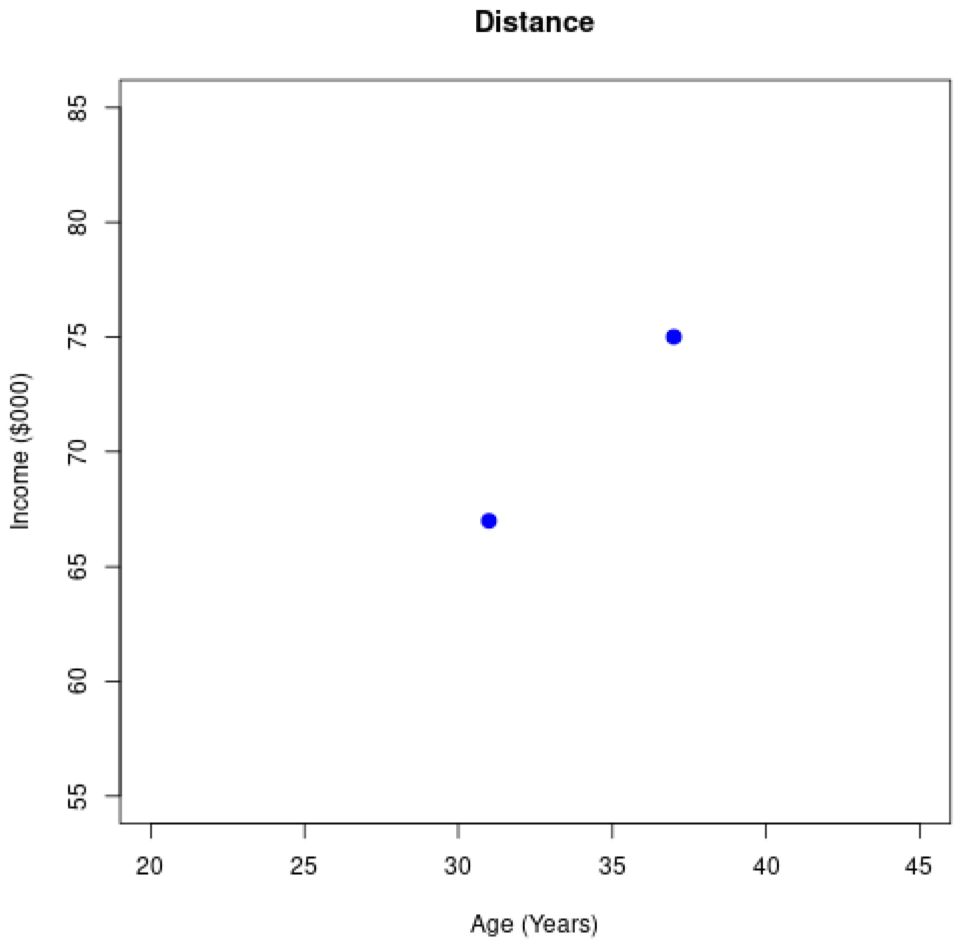
On a scatterplot



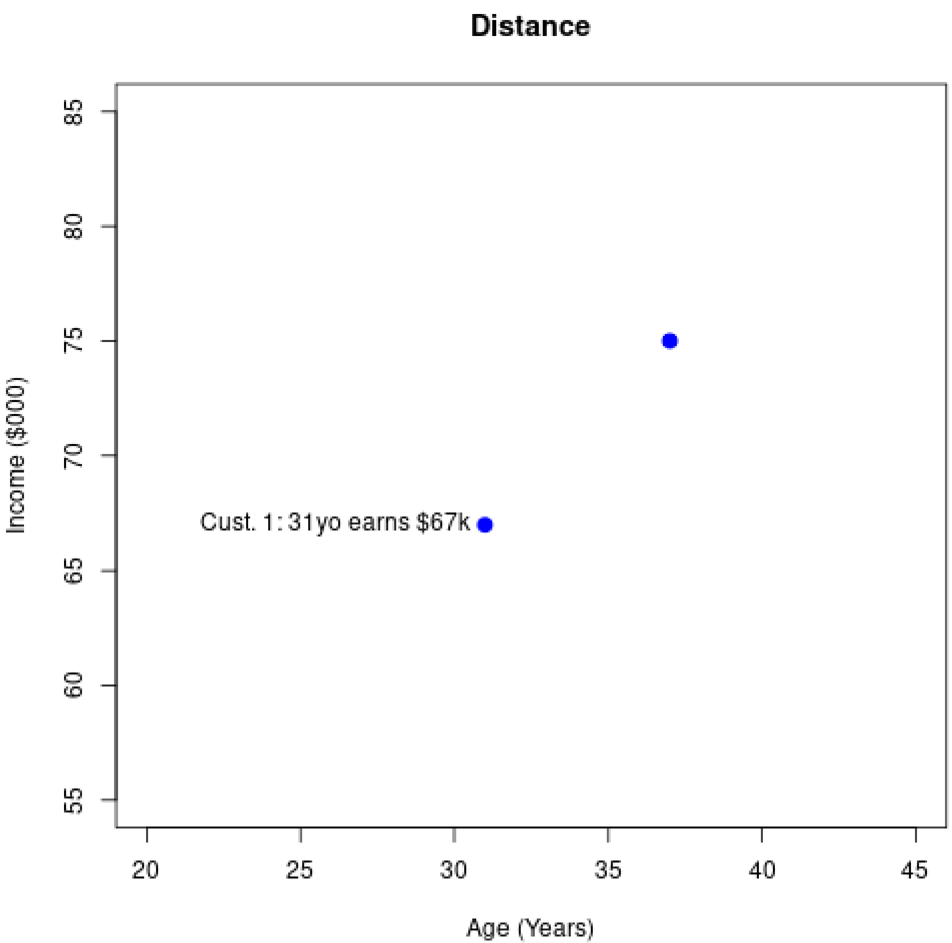
Distance as a number

- It is easy to think about three individuals but what if there are thousands of individuals?
  - In this case it will be useful to attach some number to the distance between pairs of individuals
  - We will do it with a simple application of Pythagoras' theorem.

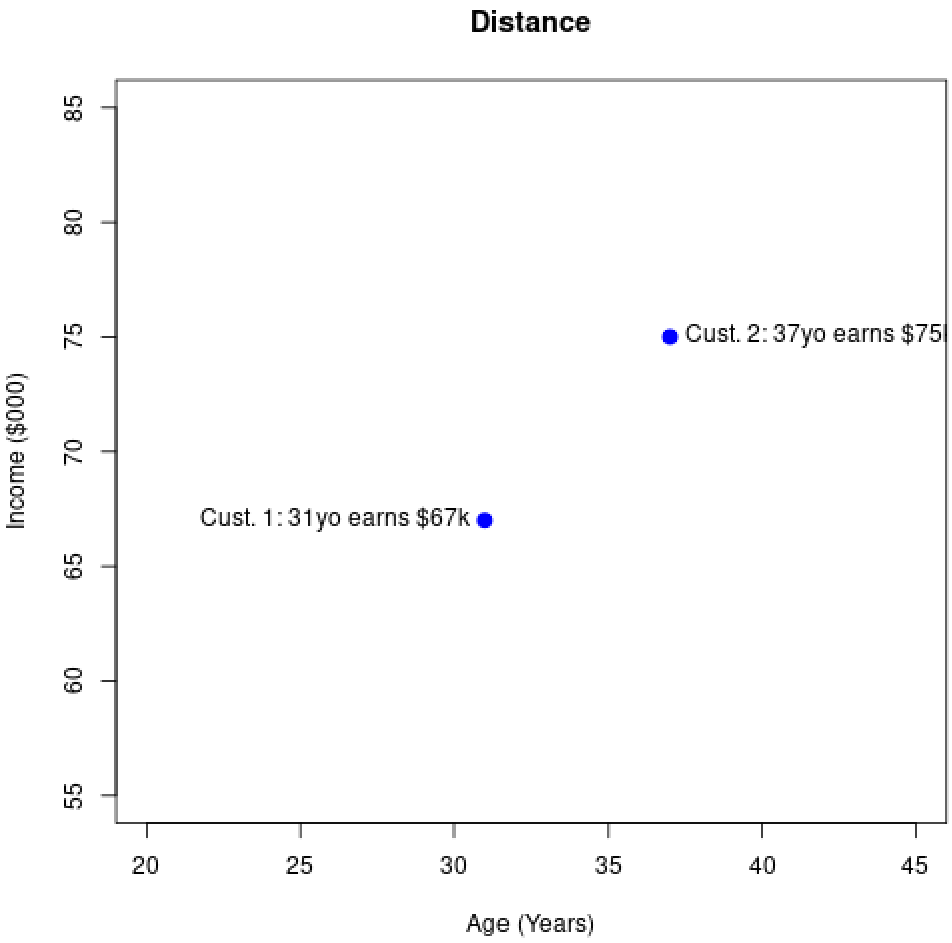
Finding the Distance



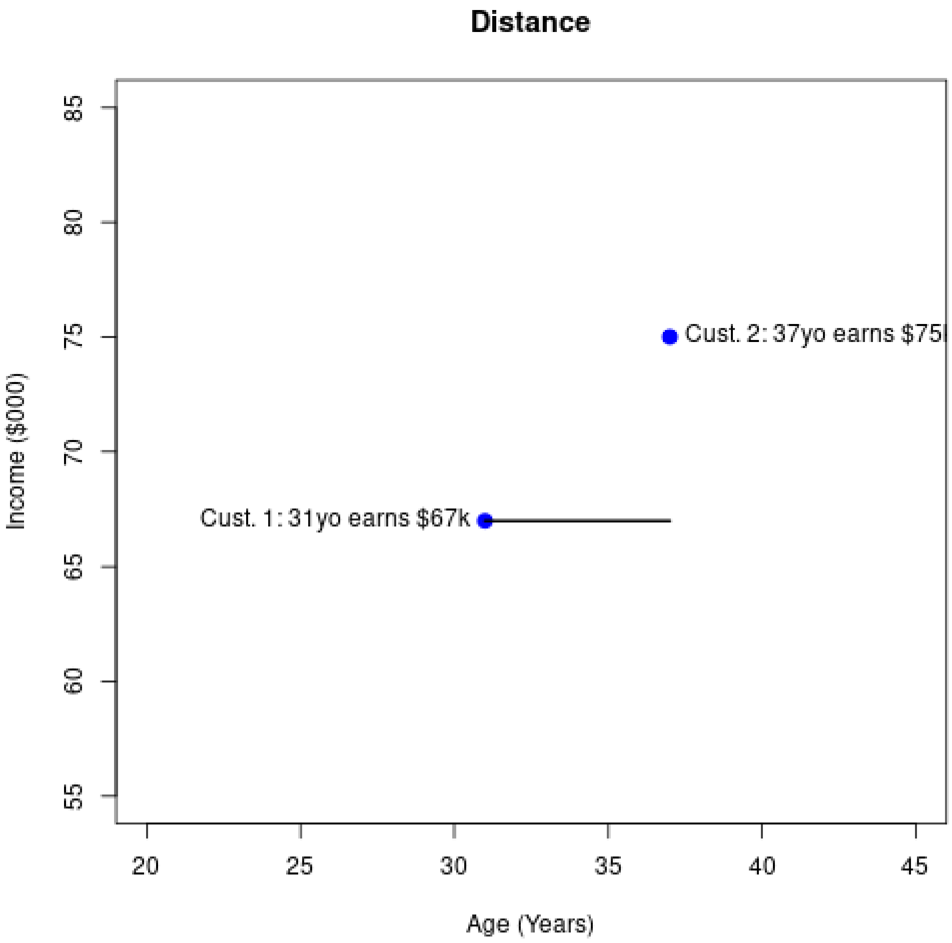
Finding the Distance



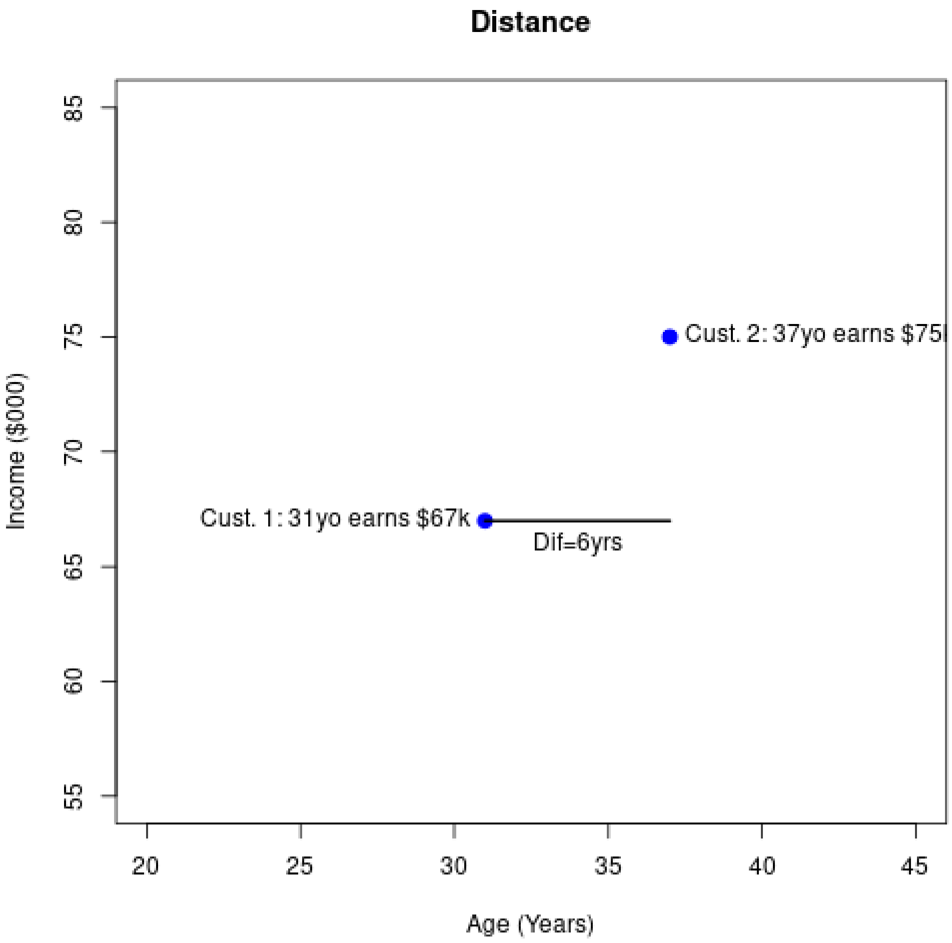
Finding the Distance



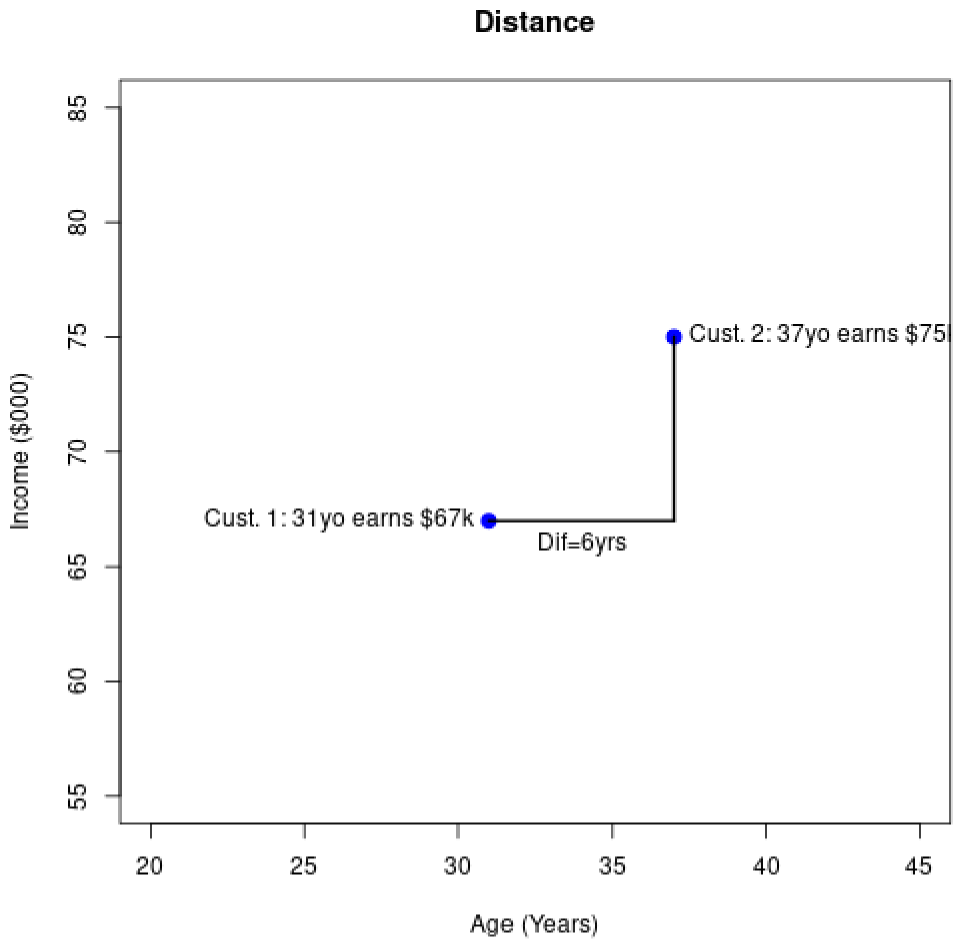
Finding the Distance



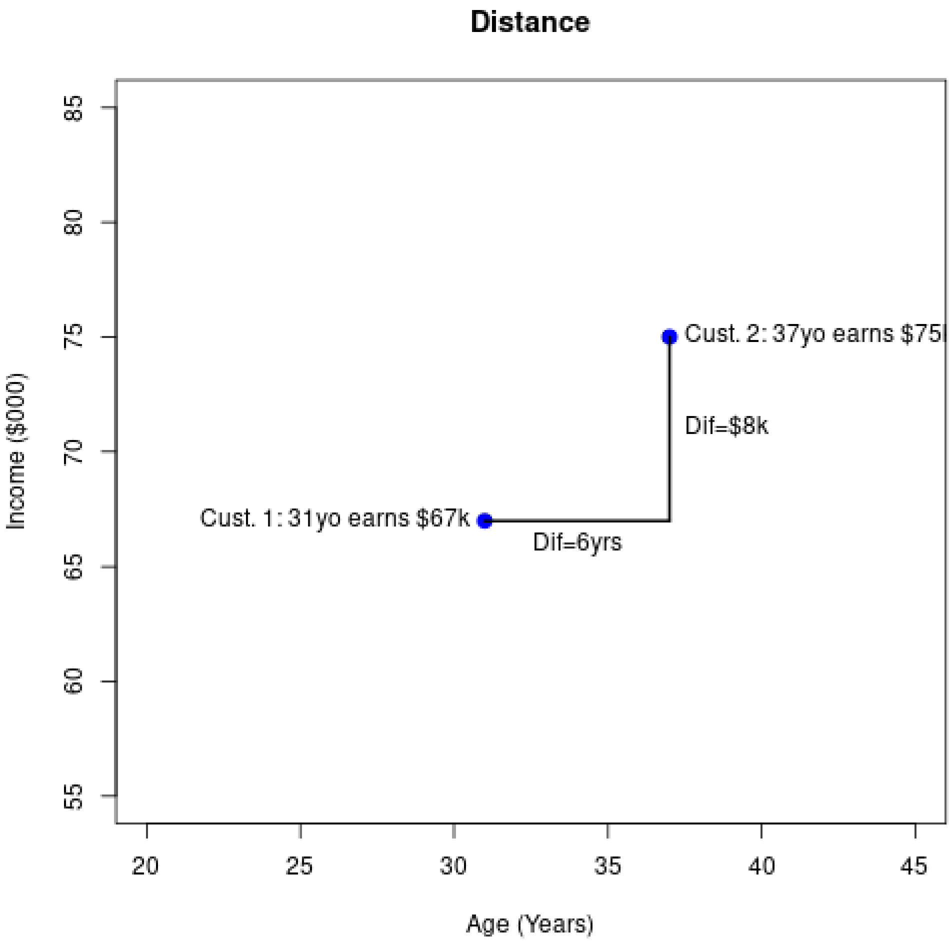
Finding the Distance



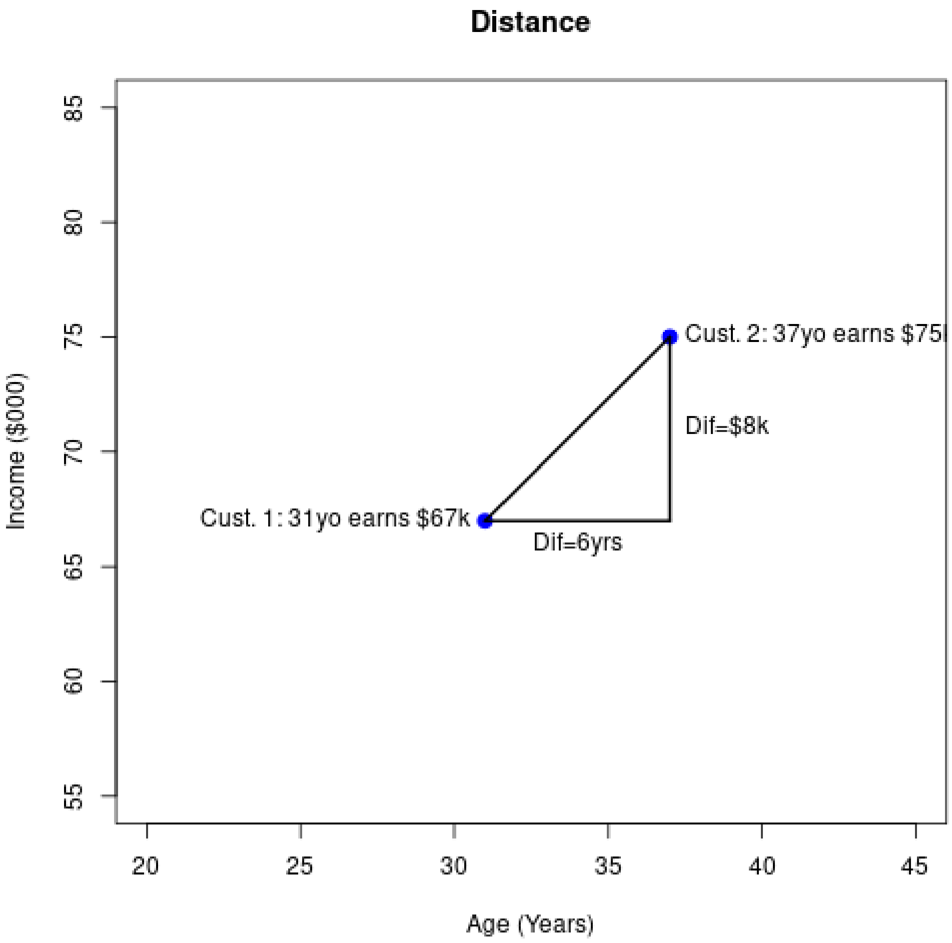
Finding the Distance



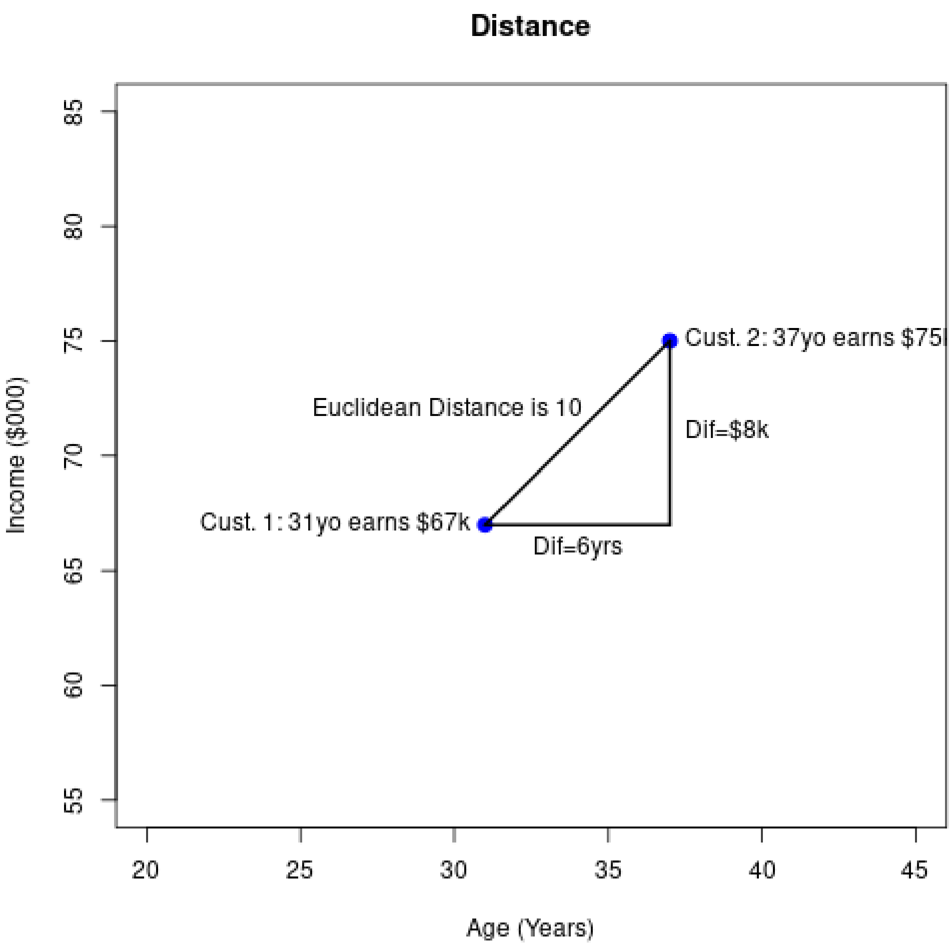
Finding the Distance



Finding the Distance



Finding the Distance



Euclidean distance

- In general there are more than two variables.
- Is there a way to apply our intuition in 2 dimensions to higher dimensions?
  - Pythagoras' theorem can be *generalised* to higher dimensions.
  - This results in a concept of distance called *Euclidean distance*.

Euclidean distance

We measure p variables for two observations: x_j is the measurement of variable j for observation $\mathbf{x}$, y_j is the measurement of variable j for observation $\mathbf{y}$.

Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ is:

$$D\left(\mathbf{x}, \mathbf{y}\right) = \sqrt{\sum_{j=1}^{p} \left(x_j - y_j\right)^2}$$

Distance and Standardising data

- We must be careful about the units of measurement.
- Euclidean distance will change when variables measured in *different units*.
- For this reason, it is common to calculate distance after the *standardising* data.
- If the variables are all measured in the same units, then this standardisation is unecessary.

Other kinds of distance

- We will nearly always use Euclidean Distance in this unit, however there are other ways of understanding distance .
- This includes distance measures for categorical data and even strings of text!
- While we will not cover these, the methods of hierarchical clustering we cover will work as long as we have some way of defining distance between individuals.

Why is distance useful?

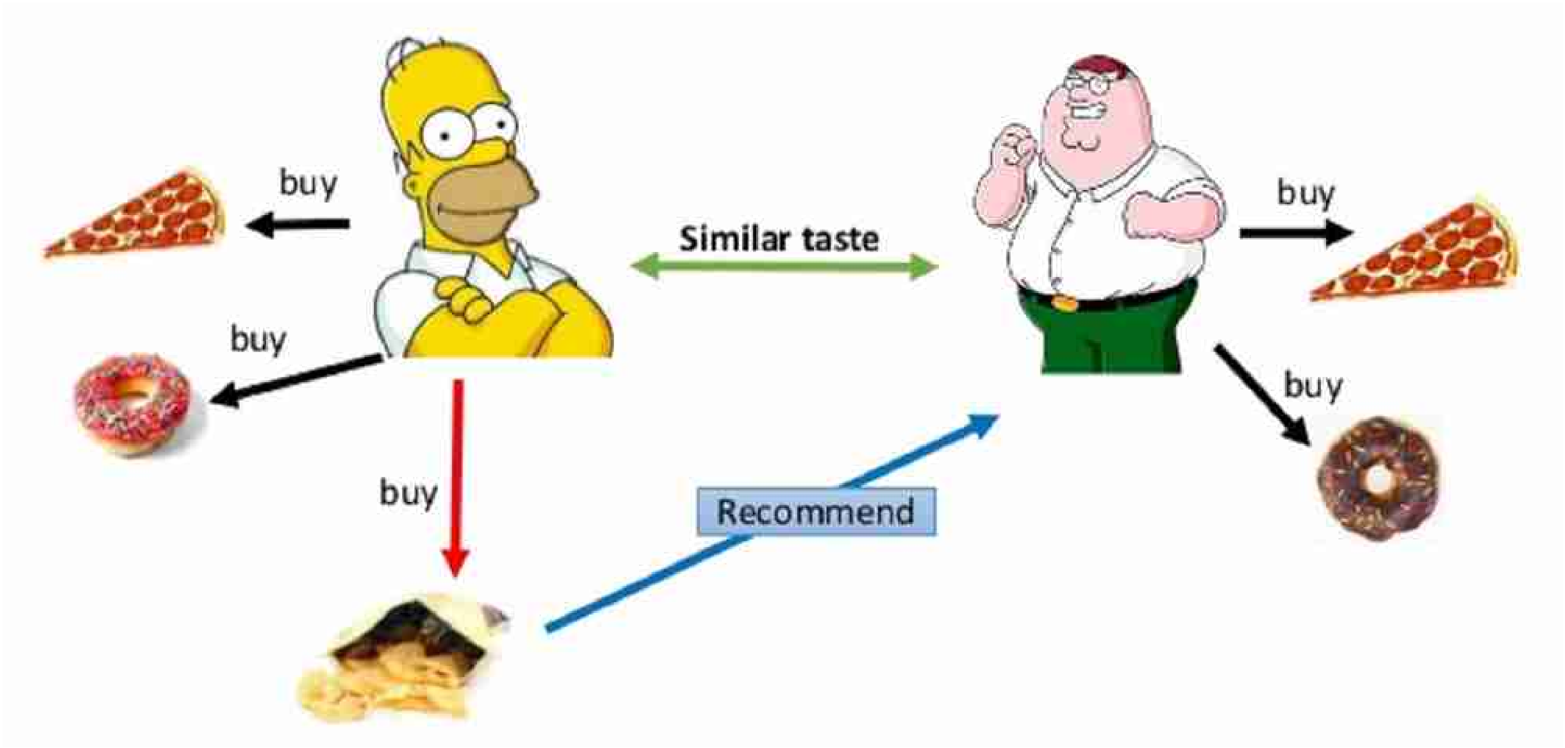


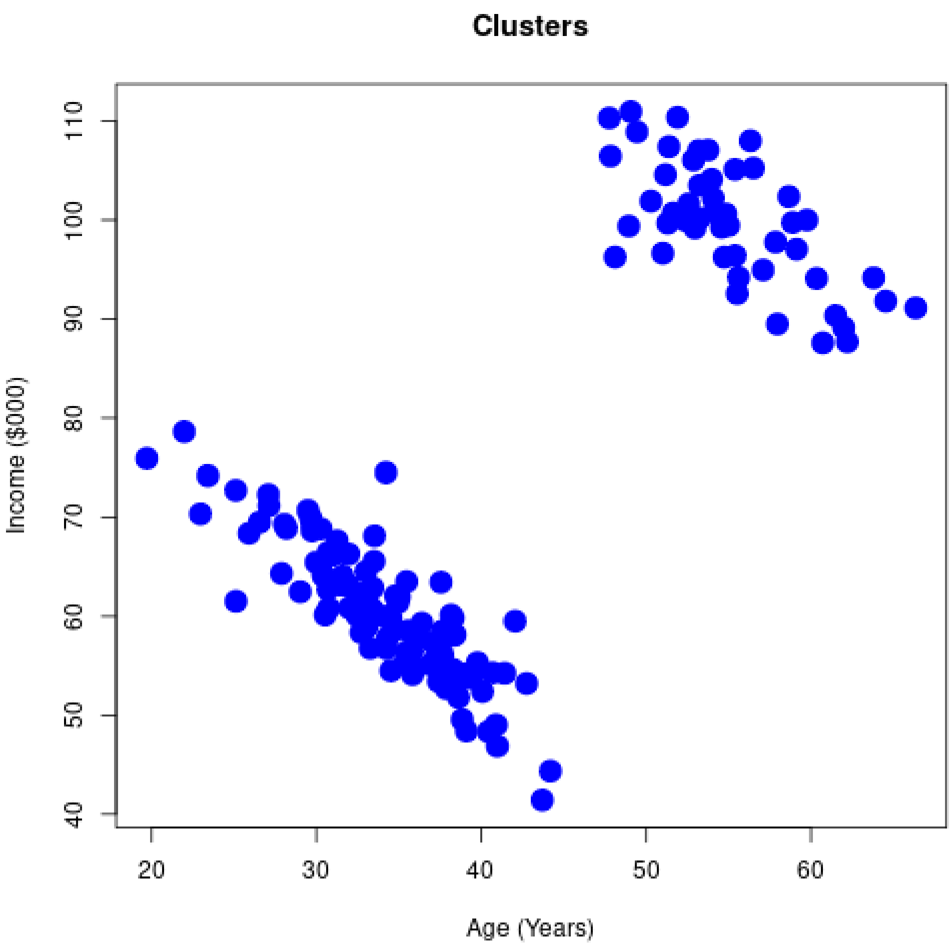
Figure by Mohamed Ben Ellefi

Recommender Systems

- Famous recommender systems are used by Amazon, Netflix, Alibaba amongst others.
- These systems are usually a hybrid of
  - Collaborative Filtering
  - Content-based Filtering
- The method we discussed is more specifically called memory-based collaborative filtering.
- Being able to put customers into similar groups is important.

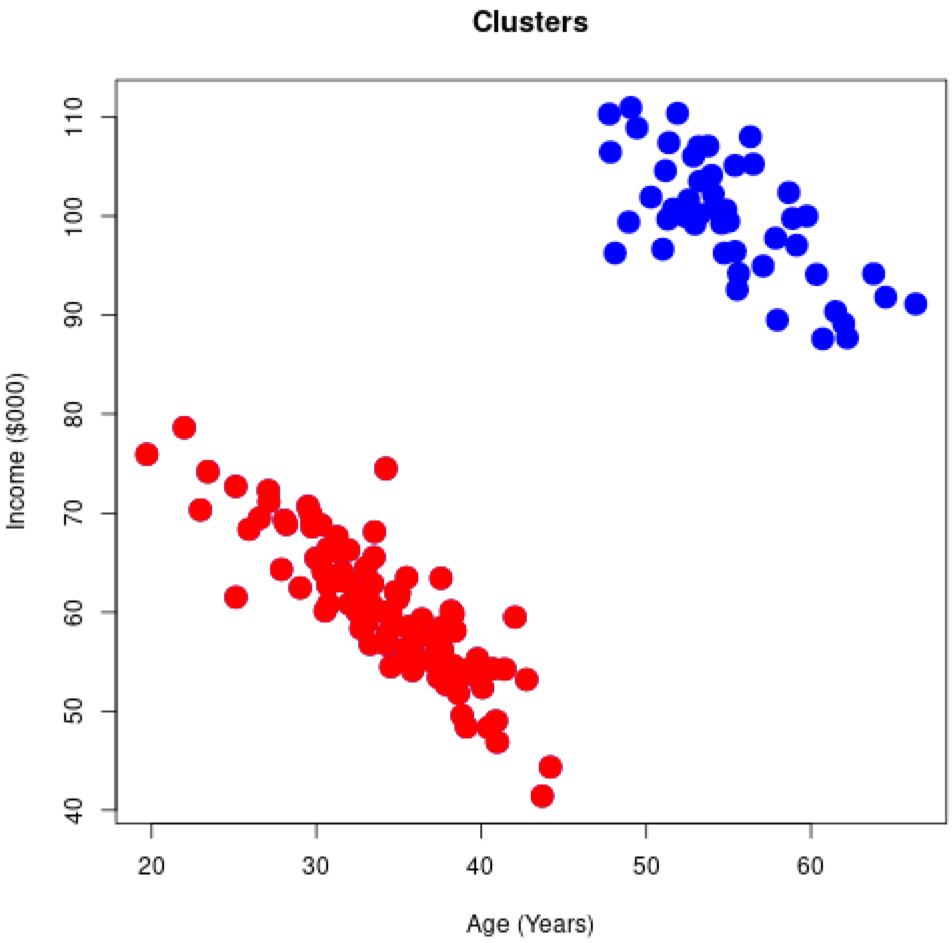
Hierarchical Clustering

Age v Income



Clusters
Income ($000)
110
100
90
80
70
60
50
40
20
30
40
50
60
Age (Years)

Obvious clusters



Summary

- When there are more than 2 variables just looking at a scatterplot doesn’t work.
- Instead algorithms can be used to find the clusters in a sensible way, even in high dimensions.

Definition of Clustering

- Oxford Dictionary: A group of similar things or people positioned or occurring closely together
- Collins Dictionary: A number of things growing, fastened, or occurring close together
- Note the importance of closeness or distance. We need two concepts of distance
  - Distance between **observations**.
  - Distance between **clusters**.

A distance between clusters

- Let $\mathcal{A}$ be a cluster with observations $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_I\}$ and $\mathcal{B}$ be a cluster with points $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_J\}$.
- The calligraphic script $\mathcal{A}$ or $\mathcal{B}$ denotes a cluster with possibly more than one point.
- The bold script $\mathbf{a}_i$ or $\mathbf{b}_j$ denotes a vector of attributes (e.g. age and income) for each observation.
- Rather than vectors, it is much easier to think of each observation as a point in a scatterplot.

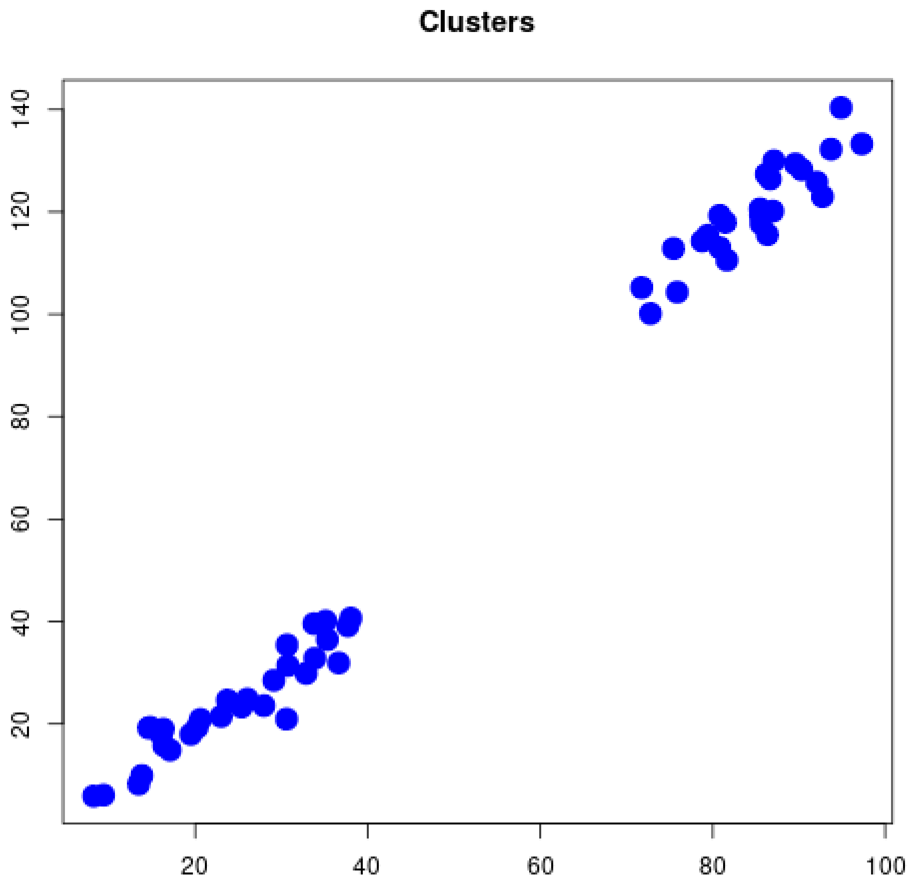
Single Linkage

One way of defining the distance between clusters $\mathcal{A}$ and $\mathcal{B}$ is

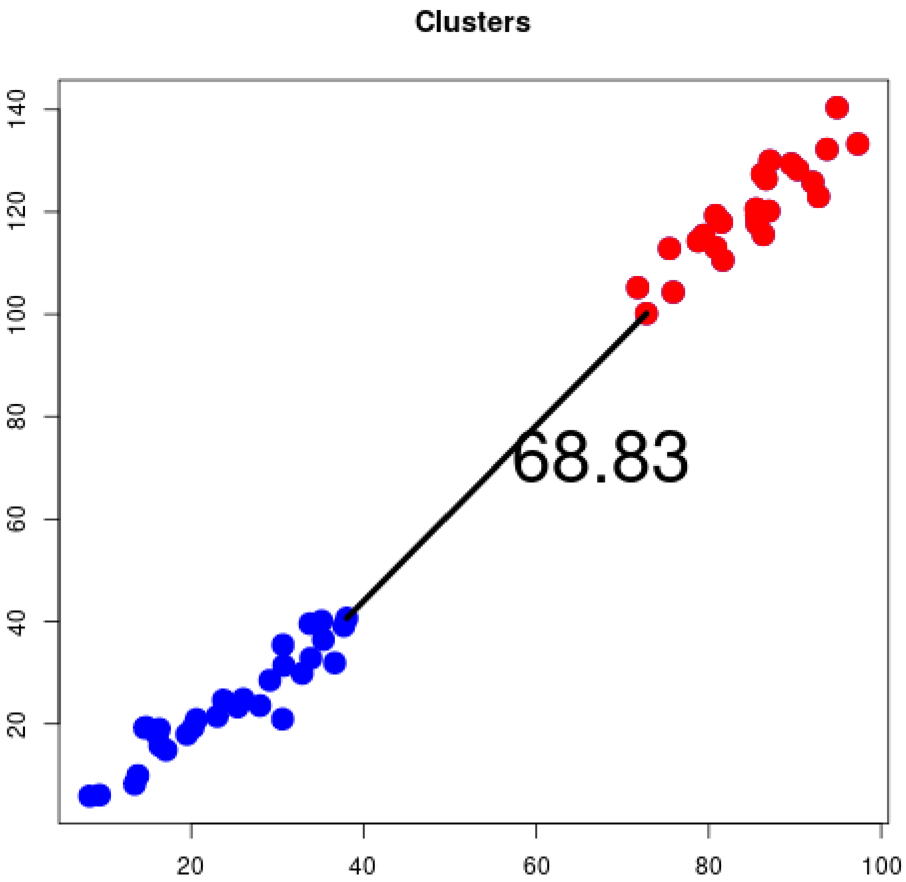
$$D(\mathcal{A}, \mathcal{B}) = \min_{i,j} D(\mathbf{a}_i, \mathbf{b}_j)$$

This is called **single linkage** or **nearest neighbour**.

Single Linkage



Single Linkage



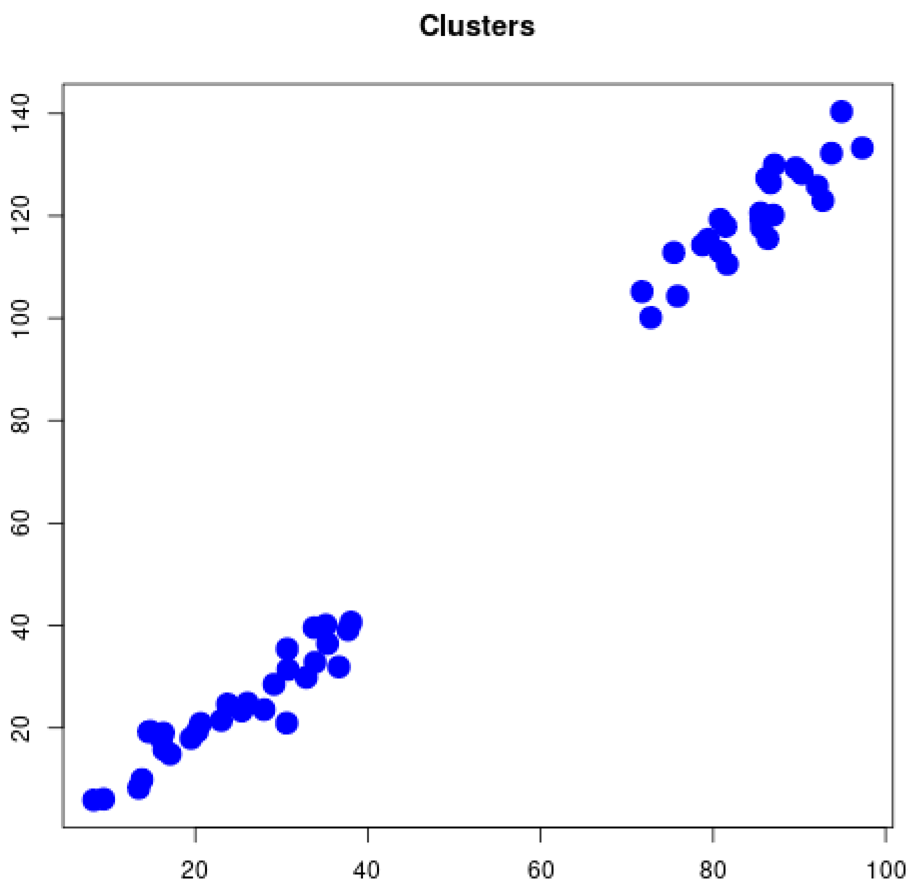
Complete Linkage

Another way of defining the distance between $\mathcal{A}$ and $\mathcal{B}$ is

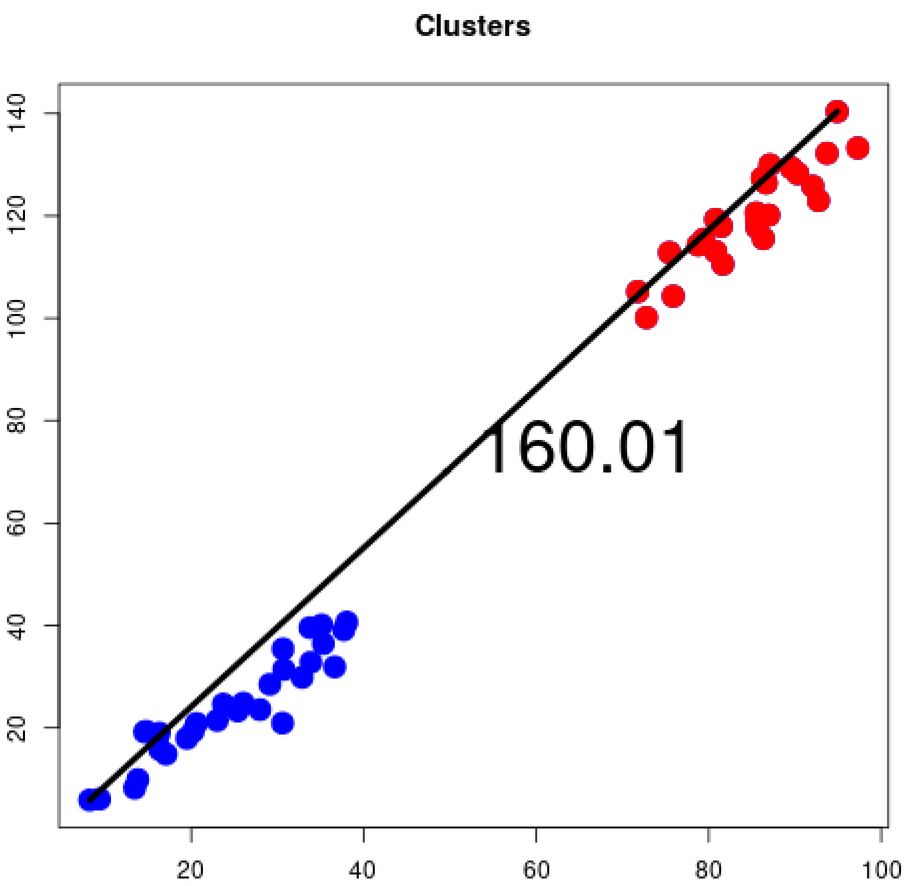
$$D(\mathcal{A}, \mathcal{B}) = \max_{i,j} D(\mathbf{a}_i, \mathbf{b}_j)$$

This is called **complete linkage** or **furthest neighbour**.

Complete Linkage



Complete Linkage



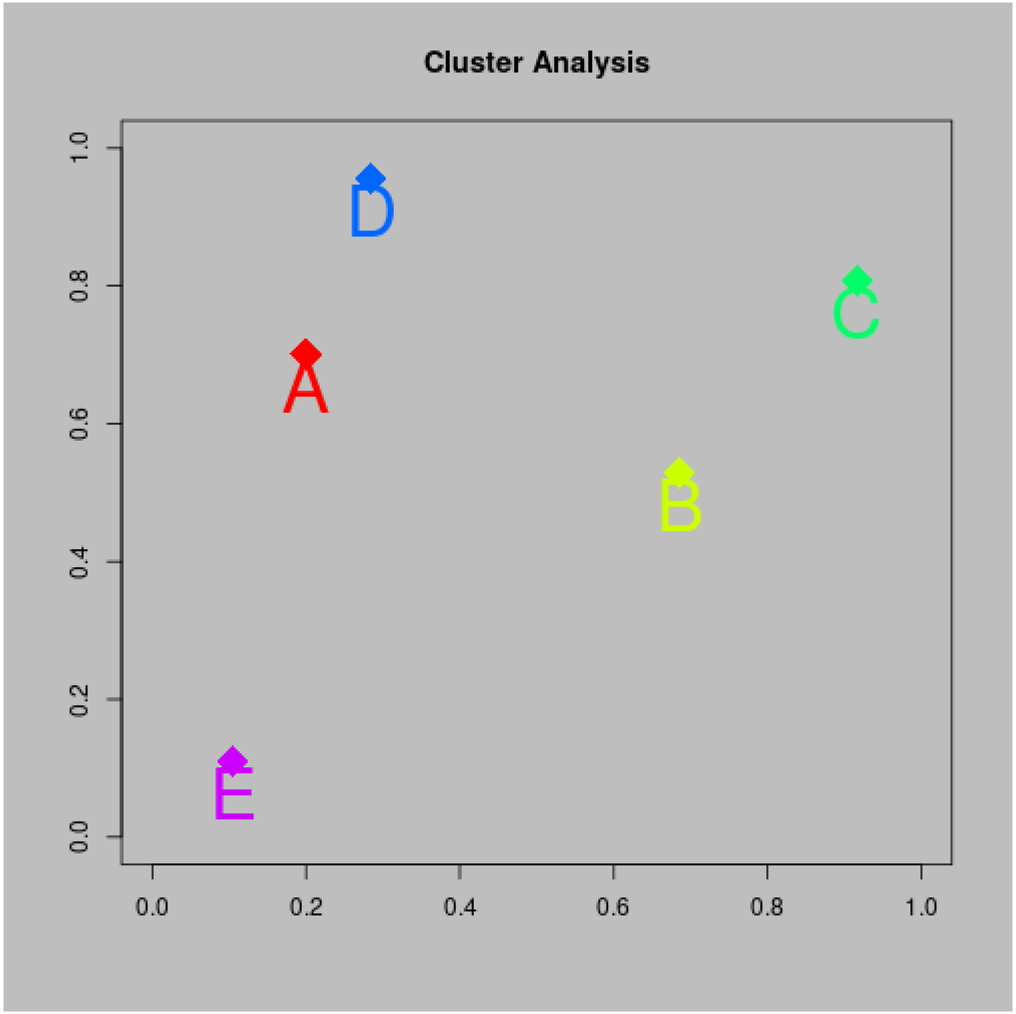
Complete linkage

- In the previous example **all** points in the red cluster are within a distance of 160.01 of **all** points in the blue cluster.
- This is why it is called **complete** linkage.

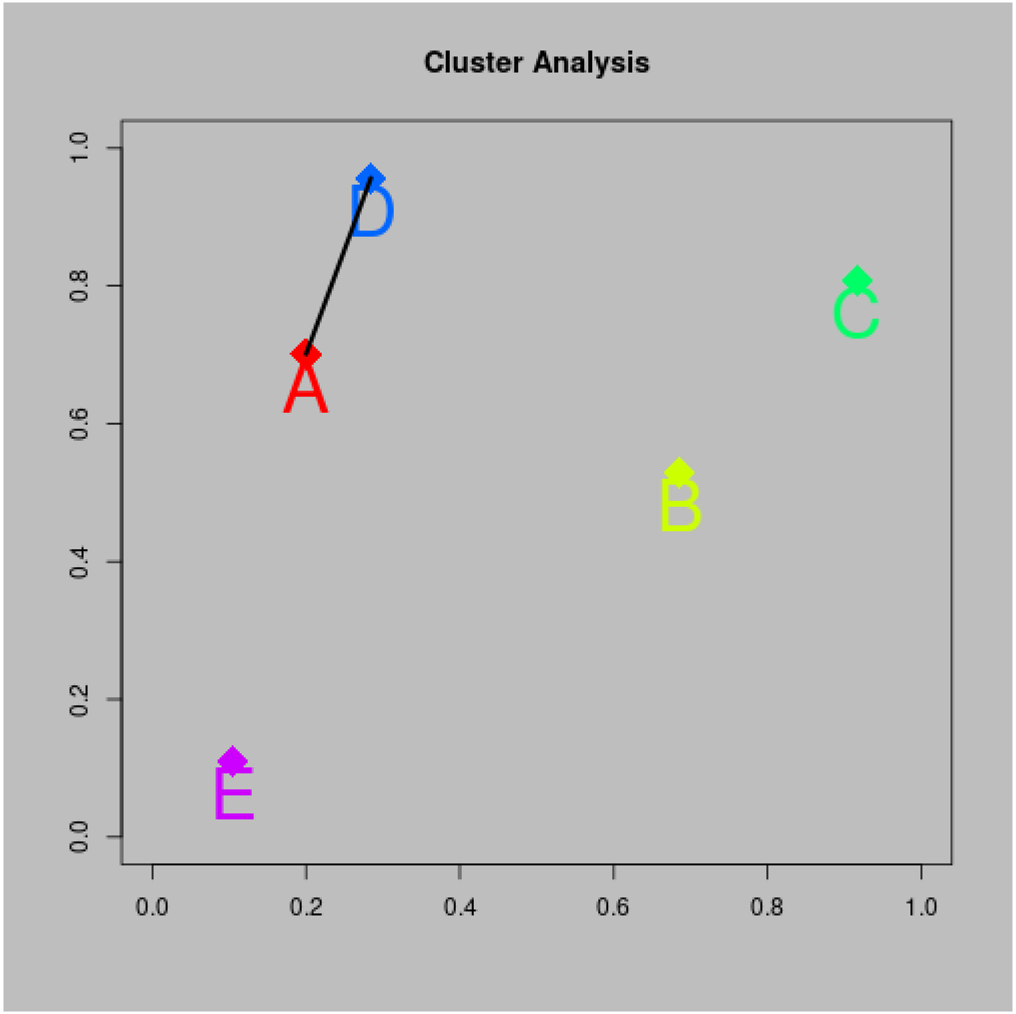
A simple example

- Over the next couple of slides we will go through the entire process of agglomerative clustering
  - We will use Euclidean distance to define distance between points
  - We will use single linkage to define the distance between clusters
- There are only five observations and two variables

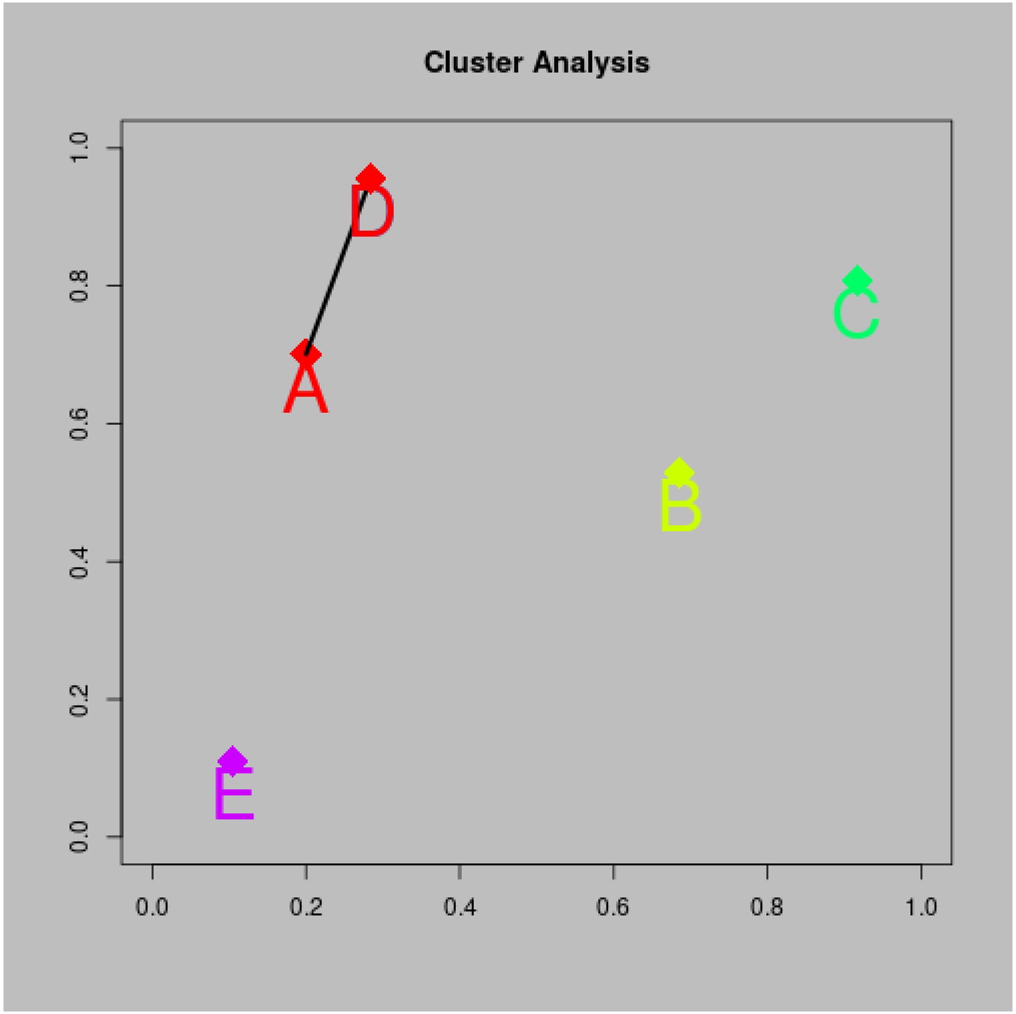
Agglomerative clustering



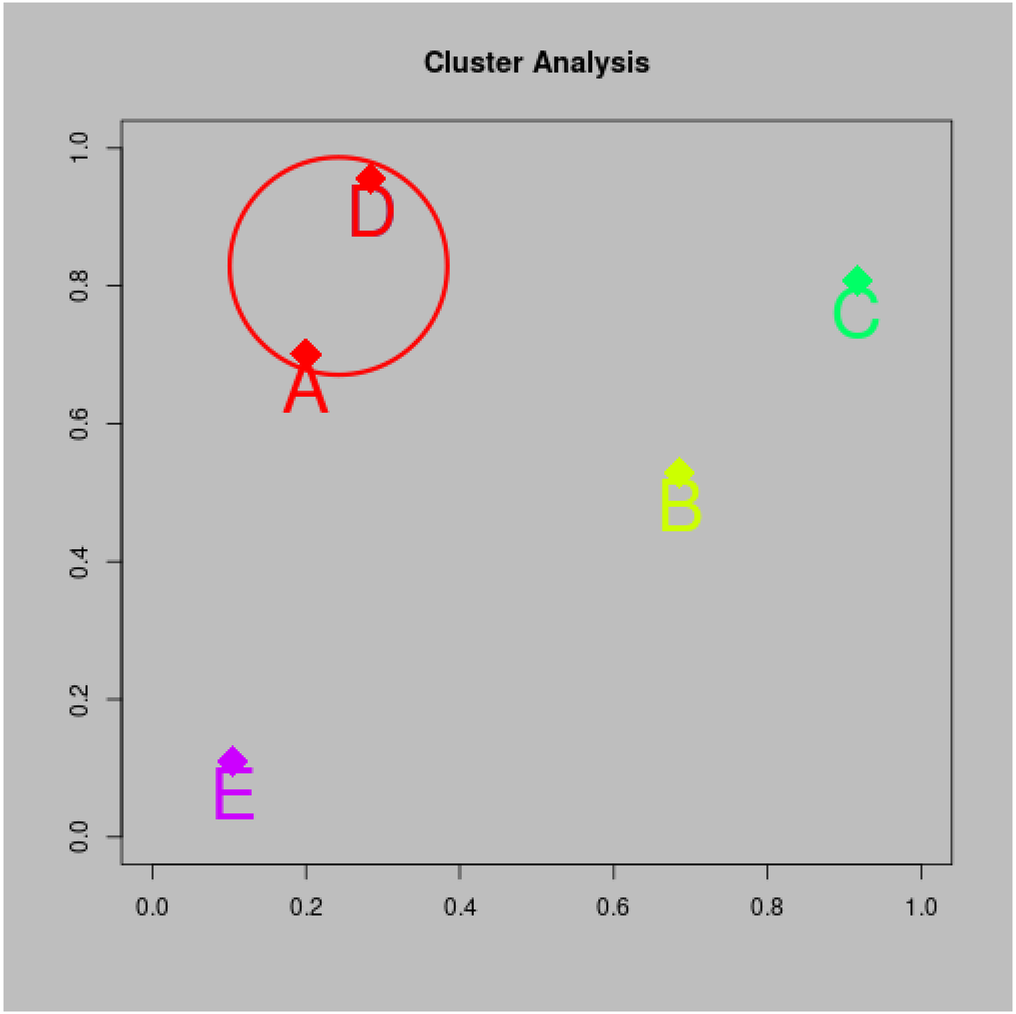
Agglomerative clustering



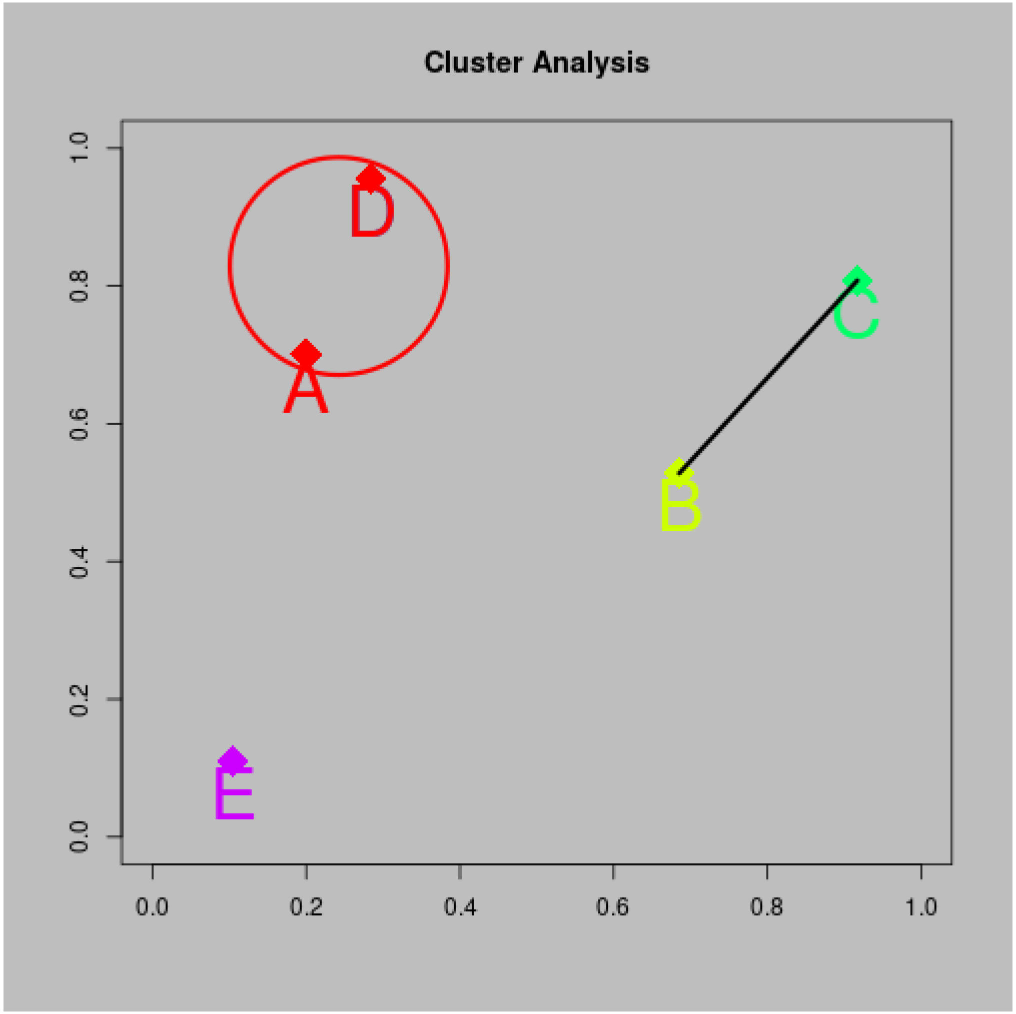
Agglomerative clustering



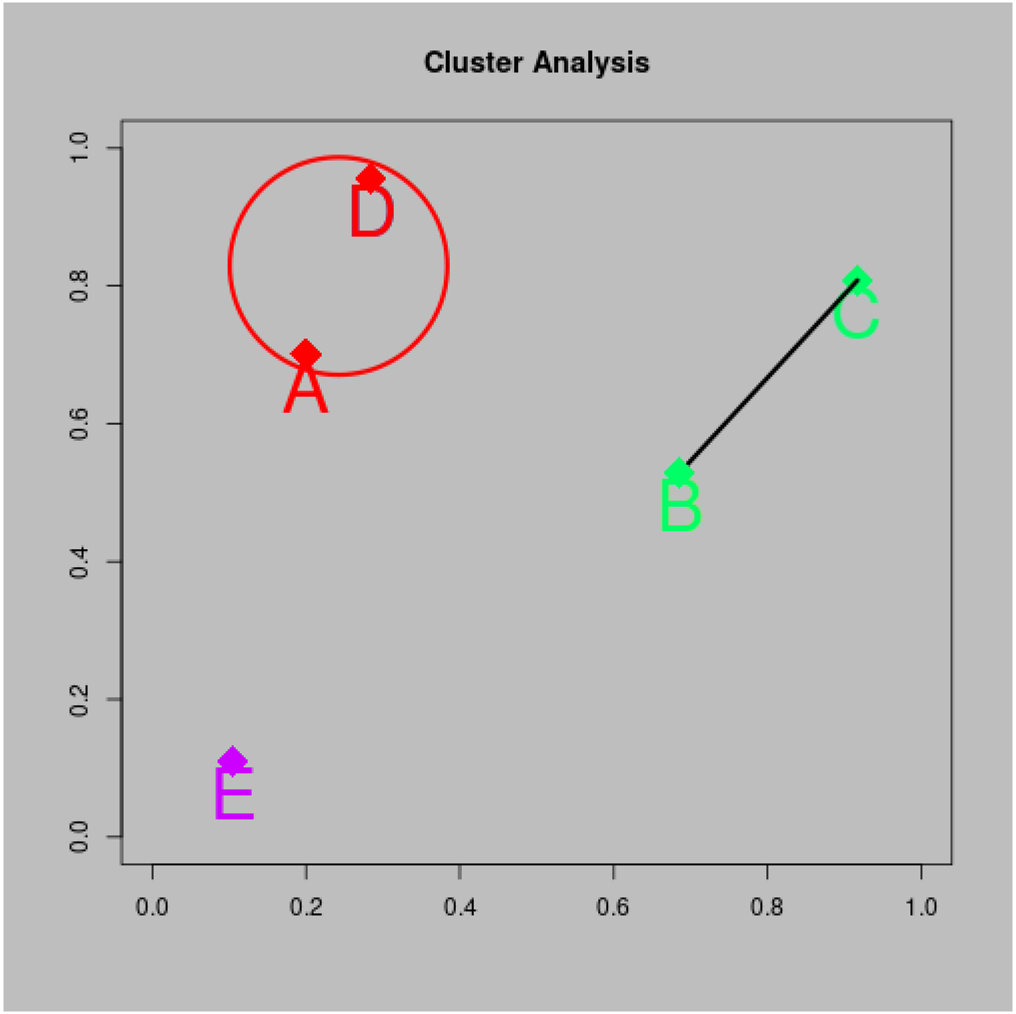
Agglomerative clustering



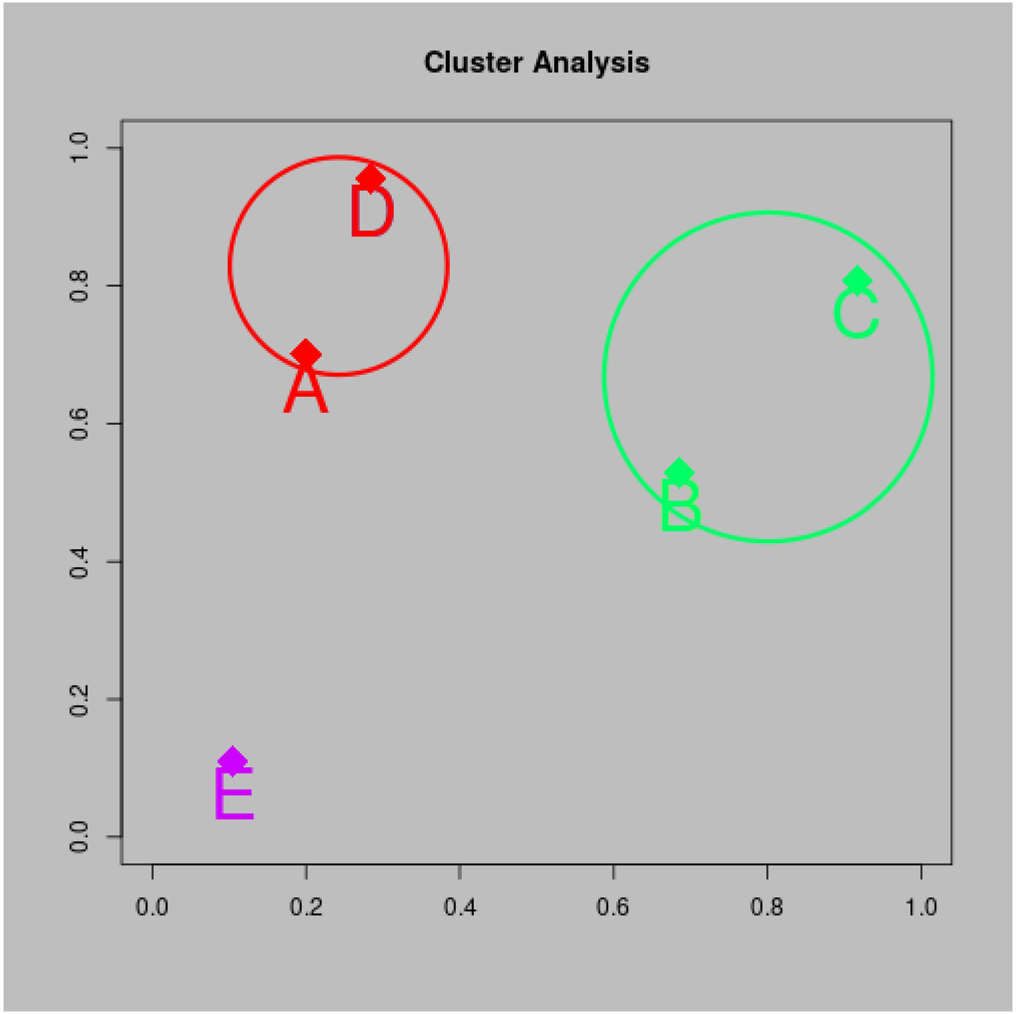
Agglomerative clustering



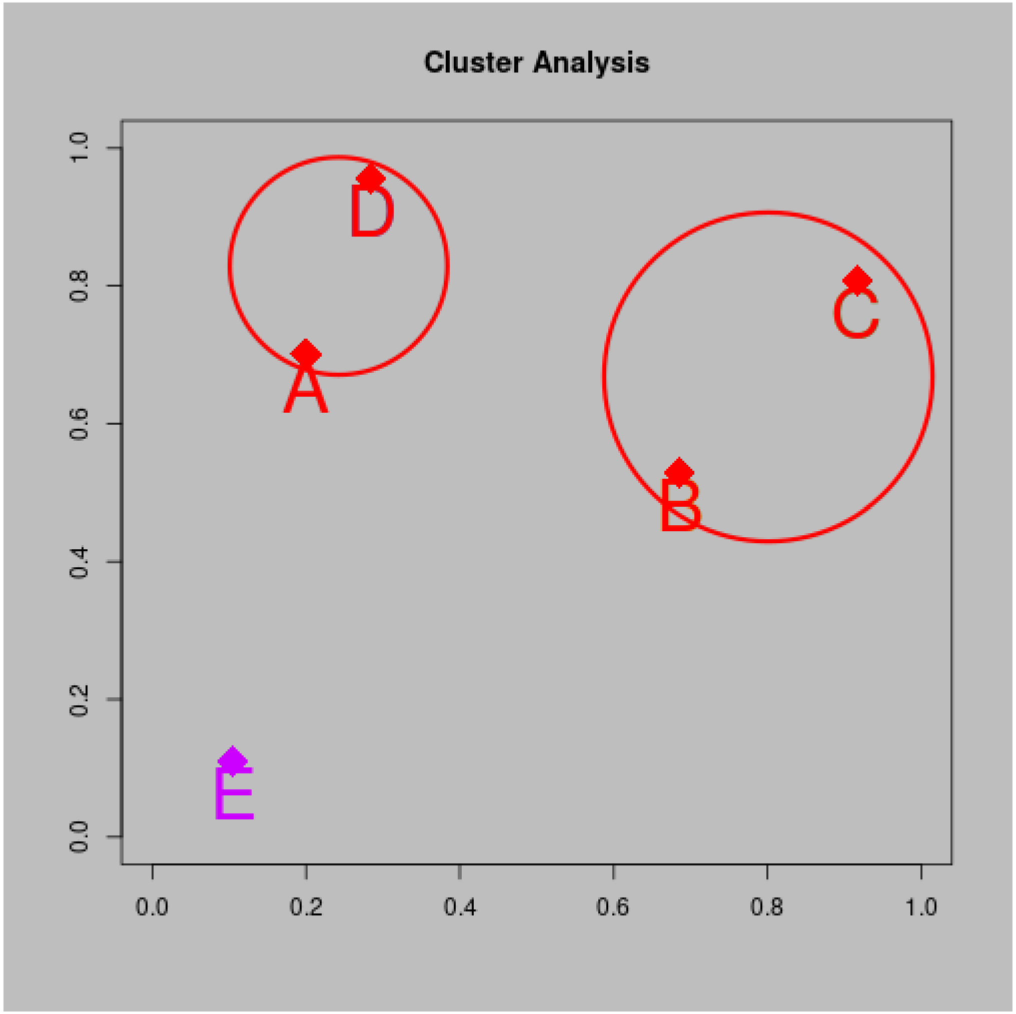
Agglomerative clustering



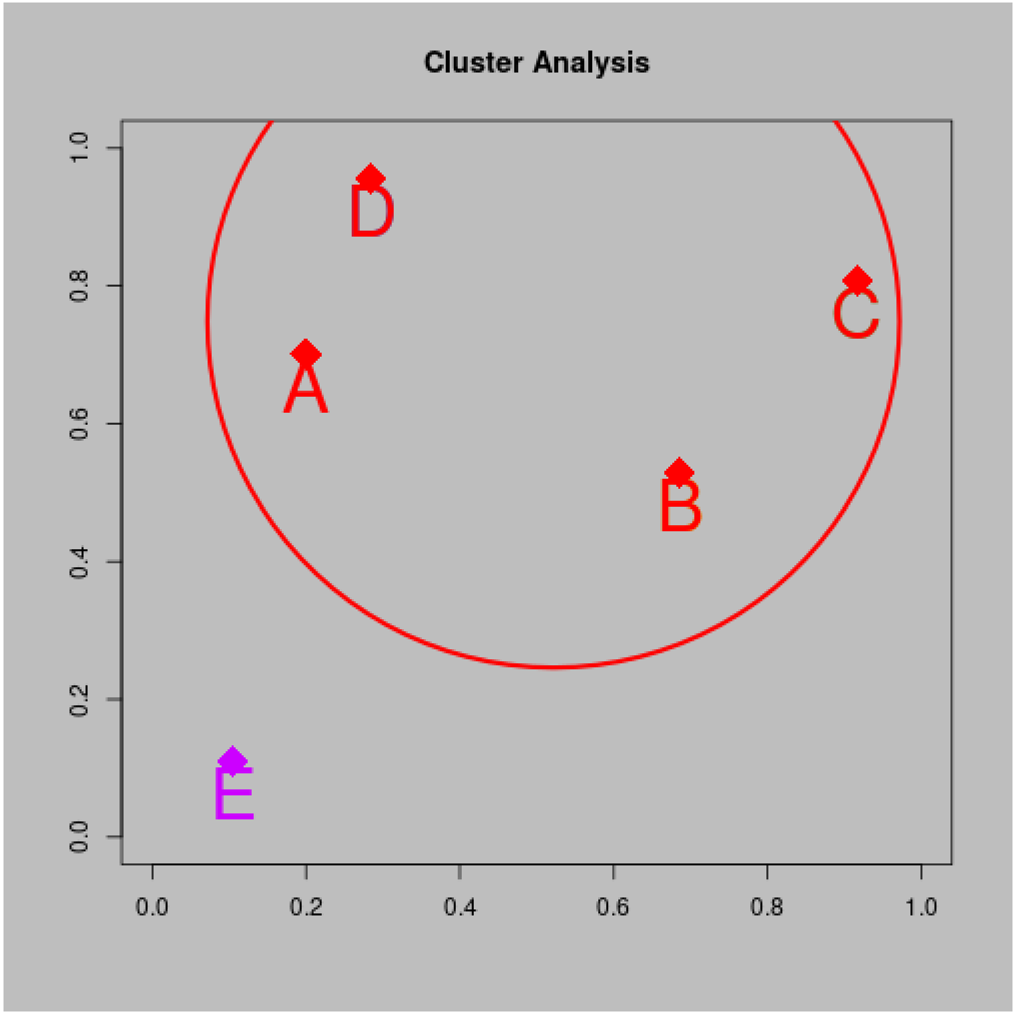
Agglomerative clustering



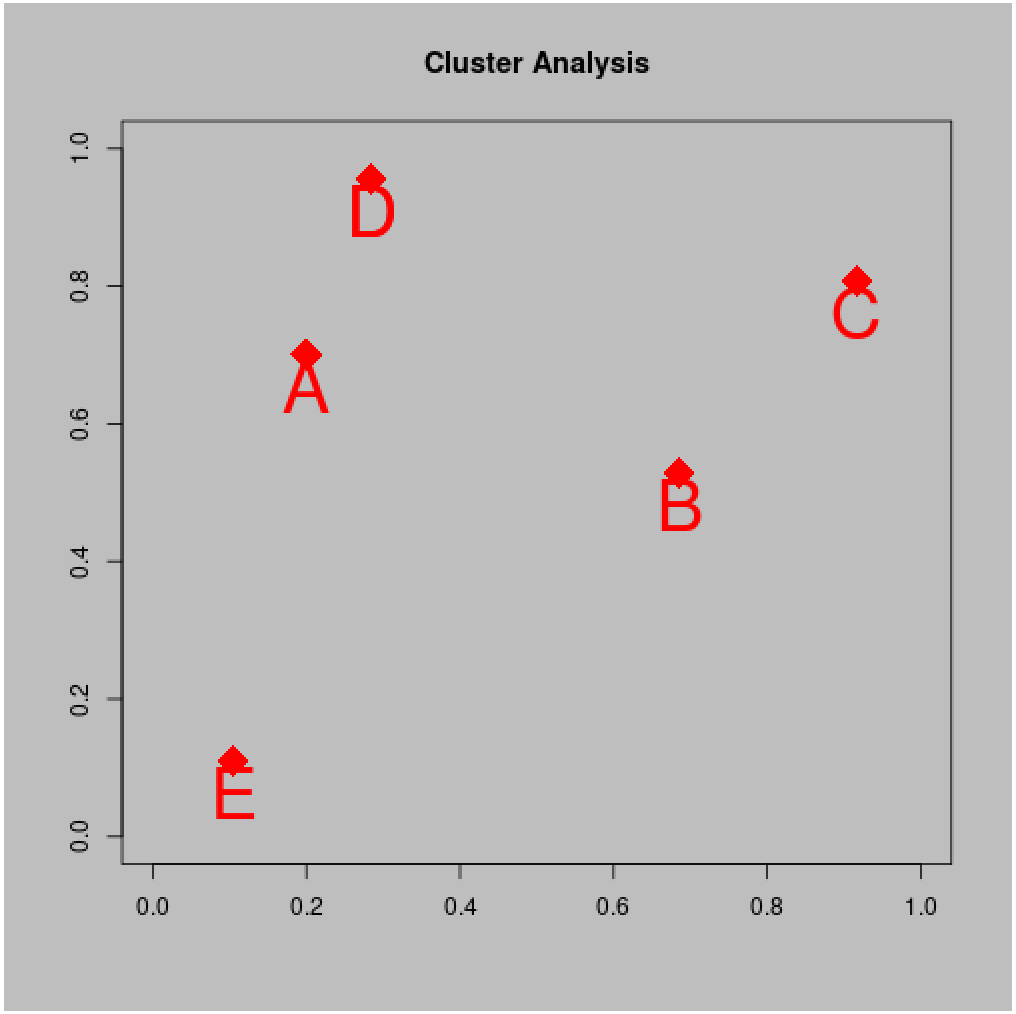
Agglomerative clustering



Agglomerative clustering



Agglomerative clustering



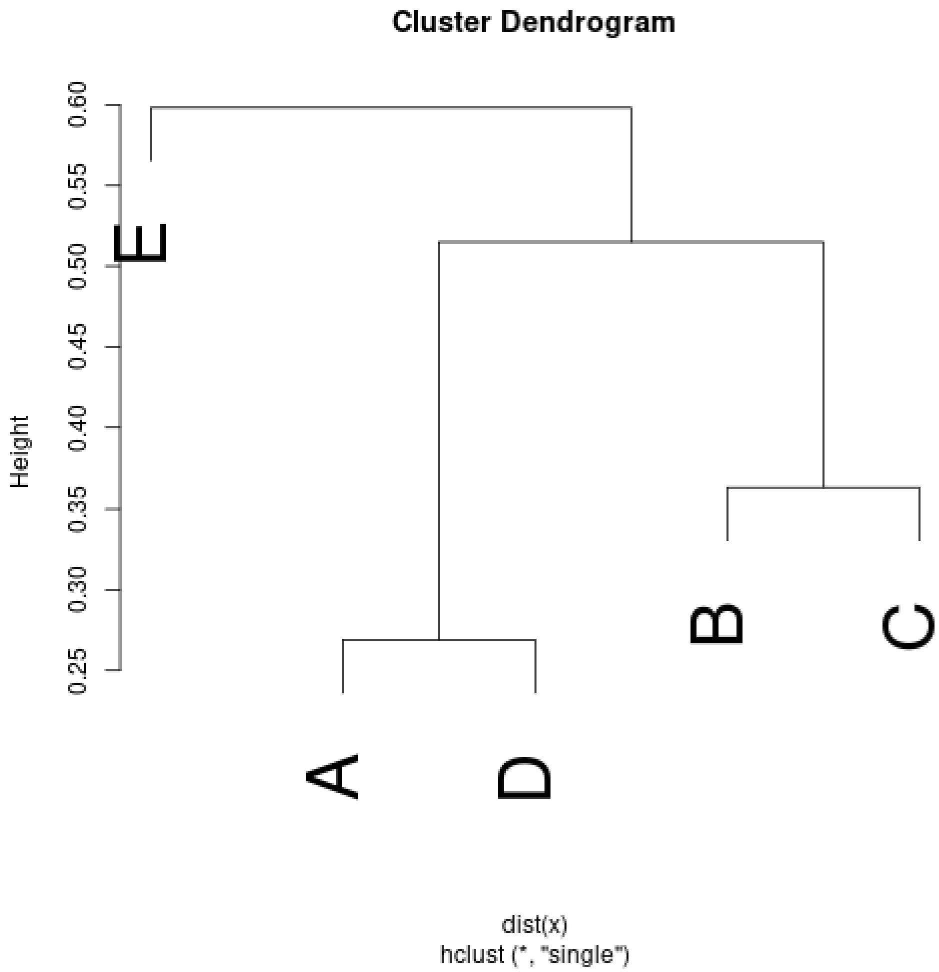
Hierarchical Clustering

- 5-cluster solution A and B and C and D and E
- 4-cluster solution {A,D} and B and C and E
- 3-cluster solution {A,D} and {B, C} and E
- 2-cluster solution {A,B, C,D} and E
- 1-cluster solution {A,B, C,D E}

Dendrogram

- The Dendrogram is a useful tool for analysing a cluster solution.
  - Observations are on one axis (usually x)
  - The distance between clusters is on other axis (usually y).
  - From the Dendrogram one can see the order in which the clusters are merged.

Dendrogram


Cluster Dendrogram
Height
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
E
A
D
B
C
dist(x)
hclust (*, "single")

Interpretation of Dendrogram

- Think of the axis with distance (y-axis) as the measuring a 'tolerance level'
- If the distance between two clusters is within the tolerance they are merged into one cluster.
- As tolerance increases more and more clusters are merged leading to less clusters overall.

A real example using Python

- We will use the mpg dataset from Seaborn
  - Observations are cars
  - Variables are related to engine size, fuel efficiency, etc.
- Will make car name the index
- Will remove non numeric variables (origin and name)
- We will drop observations with missing values.

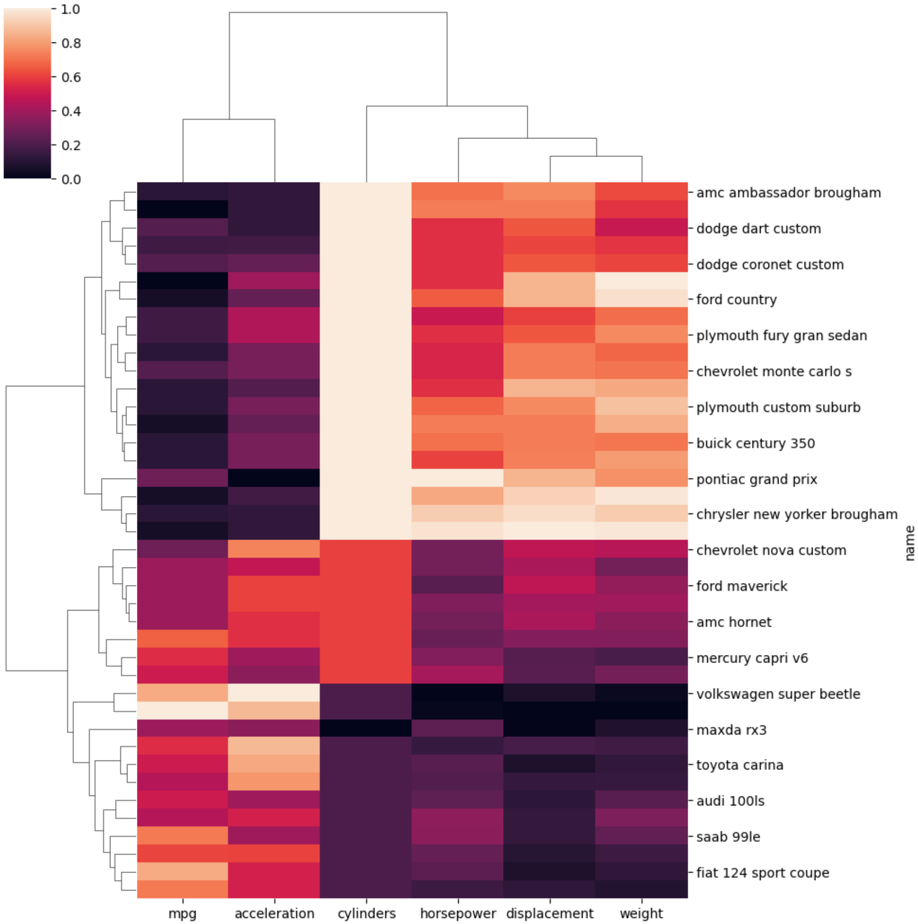
Data processing

```python
cars =  sns.load_dataset('mpg')
cars73 = cars[cars['model_year']==73]
cars73.index = cars73['name']
carsnum = cars73.iloc[:,0:6]
carsnum = carsnum.dropna(how = 'any')
carsnum
```

```
##                                 mpg  cylinders  ...  weight  acceleration
## name                                            ...
## buick century 350              13.0          8  ...    4100          13.0
## amc matador                    14.0          8  ...    3672          11.5
## chevrolet malibu               13.0          8  ...    3988          13.0
## ford gran torino               14.0          8  ...    4042          14.5
## dodge coronet custom           15.0          8  ...    3777          12.5
## mercury marquis brougham       12.0          8  ...    4952          11.5
## chevrolet caprice classic      13.0          8  ...    4464          12.0
## ford ltd                       13.0          8  ...    4363          13.0
## plymouth fury gran sedan       14.0          8  ...    4237          14.5
## chrysler new yorker brougham   13.0          8  ...    4735          11.0
## buick electra 225 custom       12.0          8  ...    4951          11.0
```

Plot

```
sns.clustermap(carsnum, standard_scale=1)
```



What do we see?

- Notice there are two dendrograms
  - One groups observations together
  - The other groups variables together
- The inside is a heatmap for the data matrix
- Cars most easily grouped by cylinders.
- Also groupings in variables.

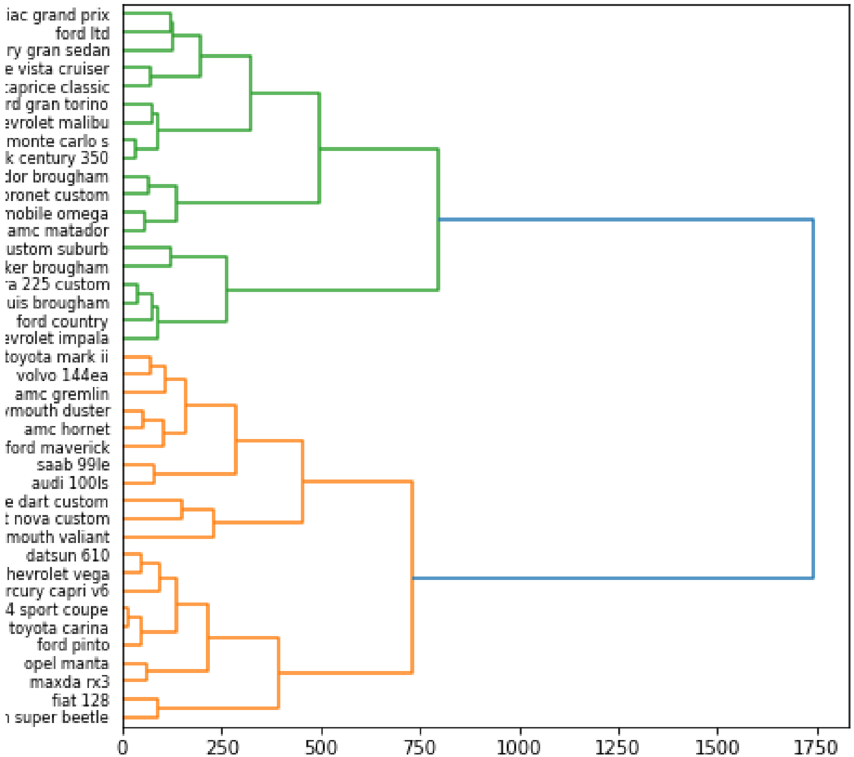
Dendrogram only

```
from scipy.cluster.hierarchy import dendrogram, linkage
import numpy as np
plt.figure()
Z = linkage(carsnum, 'average')
dendrogram(Z, orientation = 'right', leaf_font_size=8, labels=carsnum.i
```

```
## {'icoord': [[5.0, 5.0, 15.0, 15.0], [25.0, 25.0, 35.0, 35.0], [55.0, 55.0,
```

Dendrogram

```
plt.show()
```



More about the code

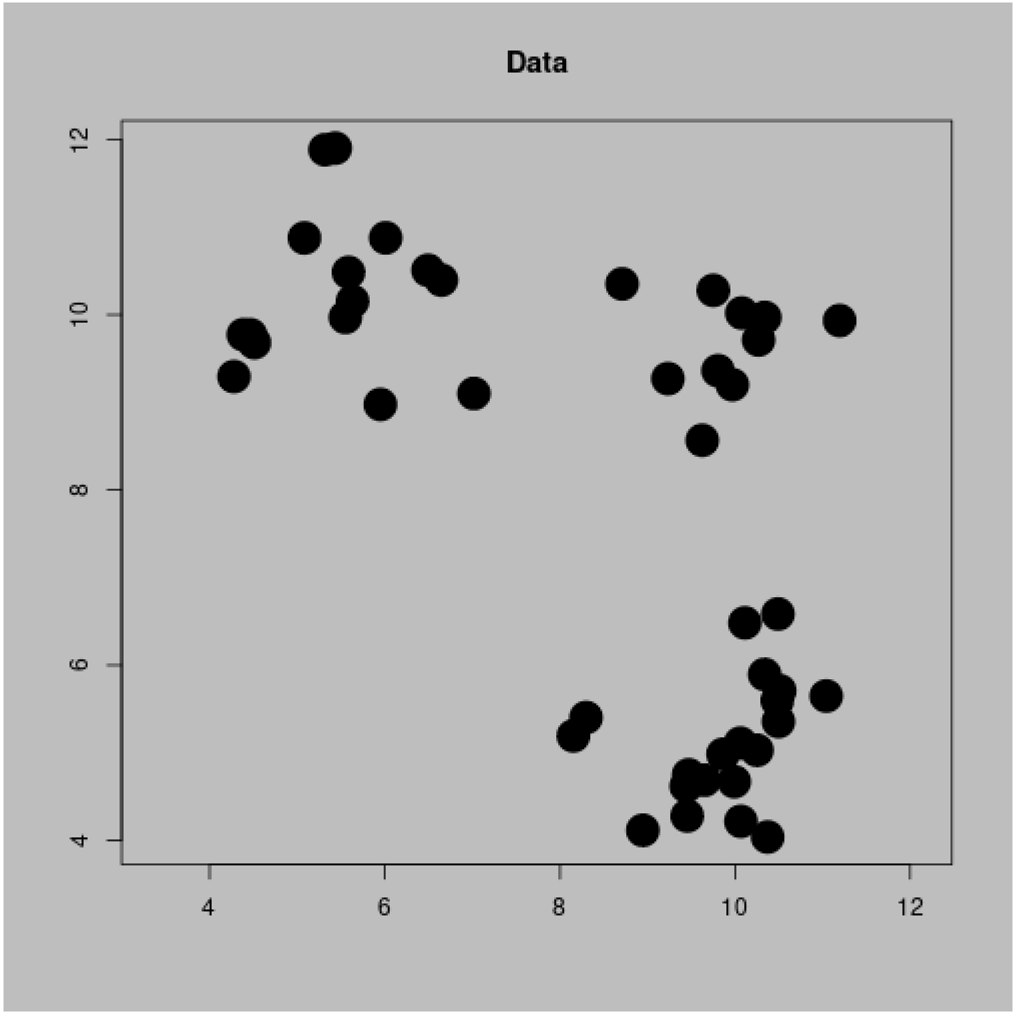
- The hierarchical clustering is done using the scipy package.
- Information can also be pulled out of the object created by this package.
- By default this package does not do simple linkage.
- We will now see why.

Other clustering methods

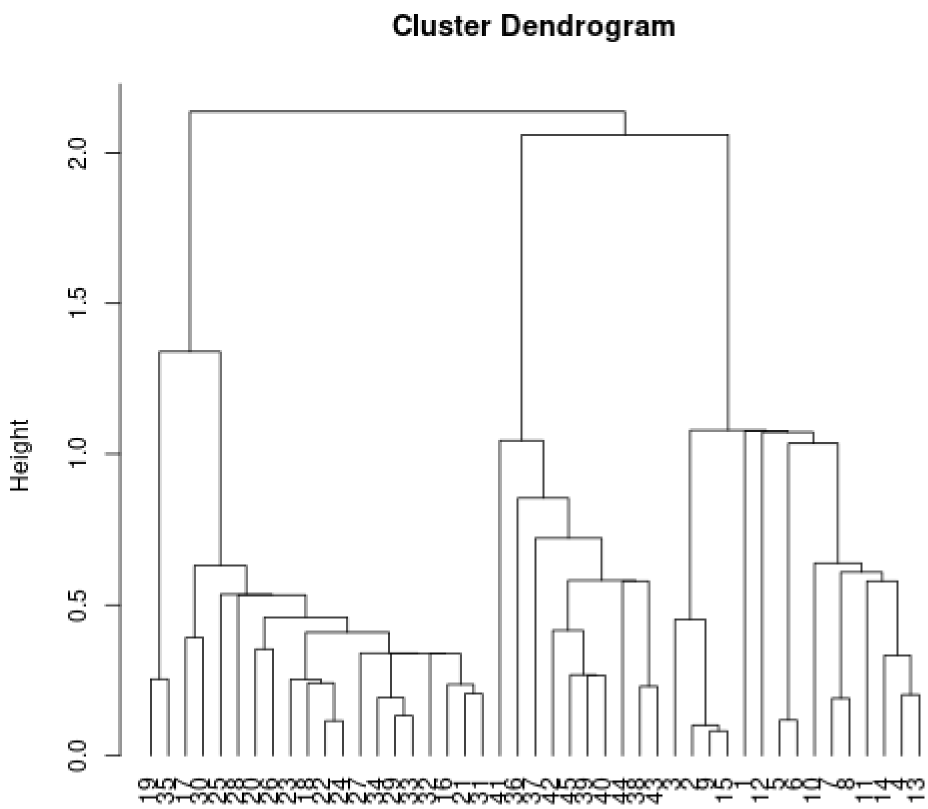
Pros and Cons of Single Linkage

- Pros:
  - Single linkage is very easy to understand.
  - Single linkage is a very fast algorithm.
- Cons:
  - Single linkage is very sensitive to single observations which leads to chaining.
  - Complete linkage avoids this problem and gives more compact clusters with a similar diameter.

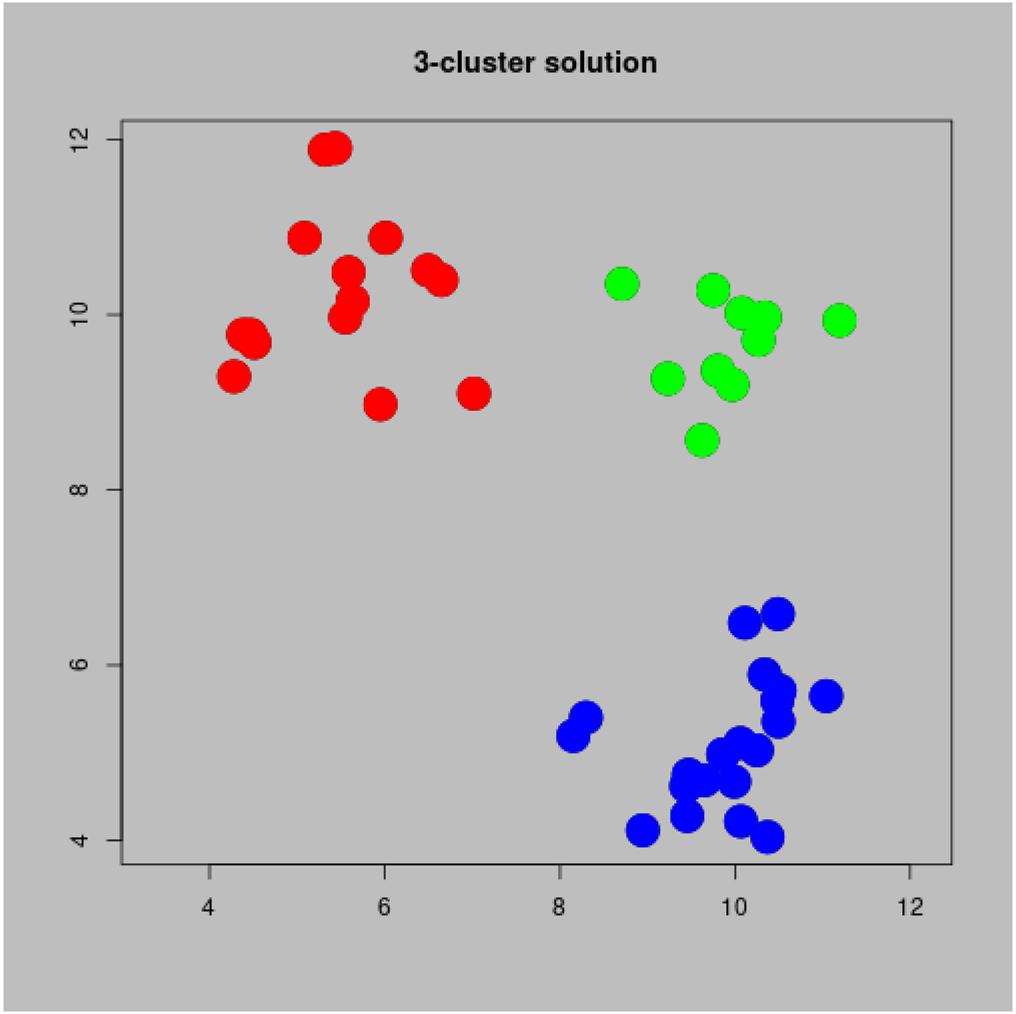
Chaining



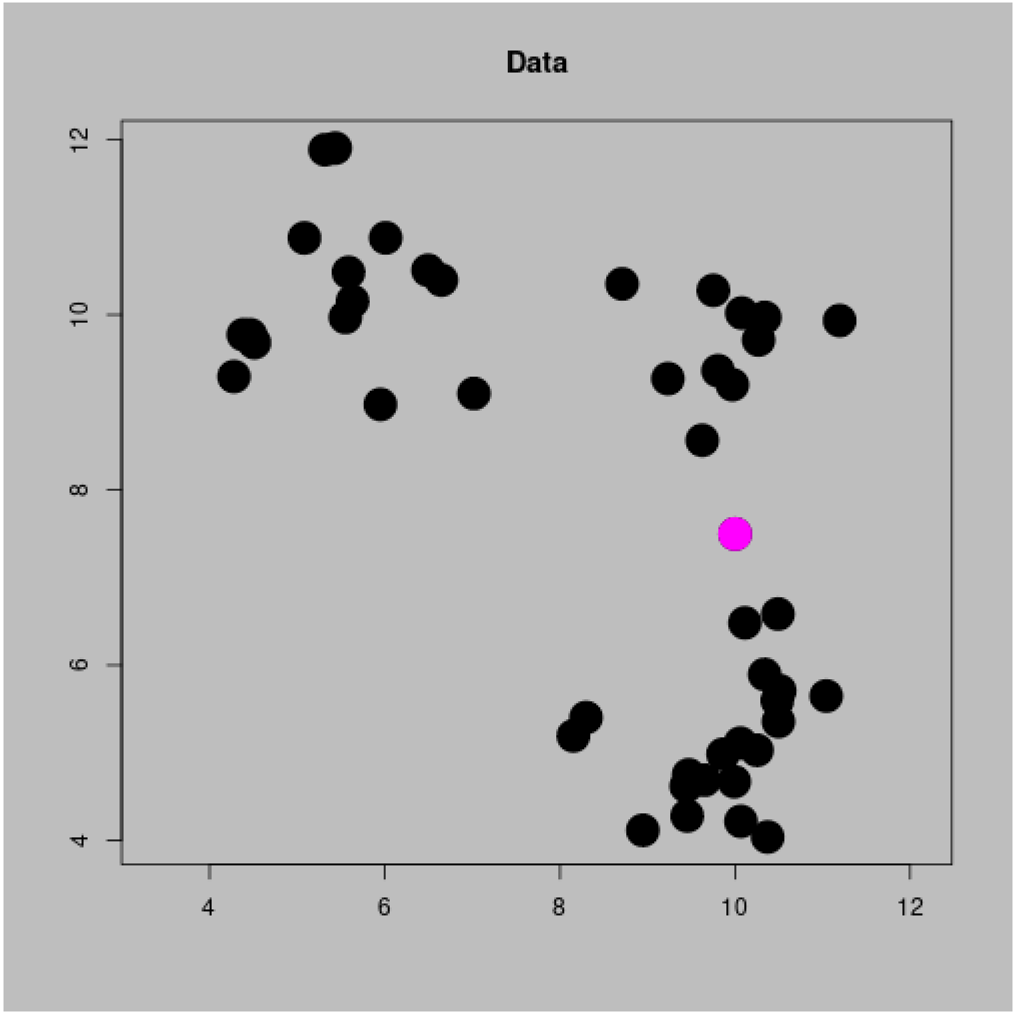
Single Linkage Dendrogram



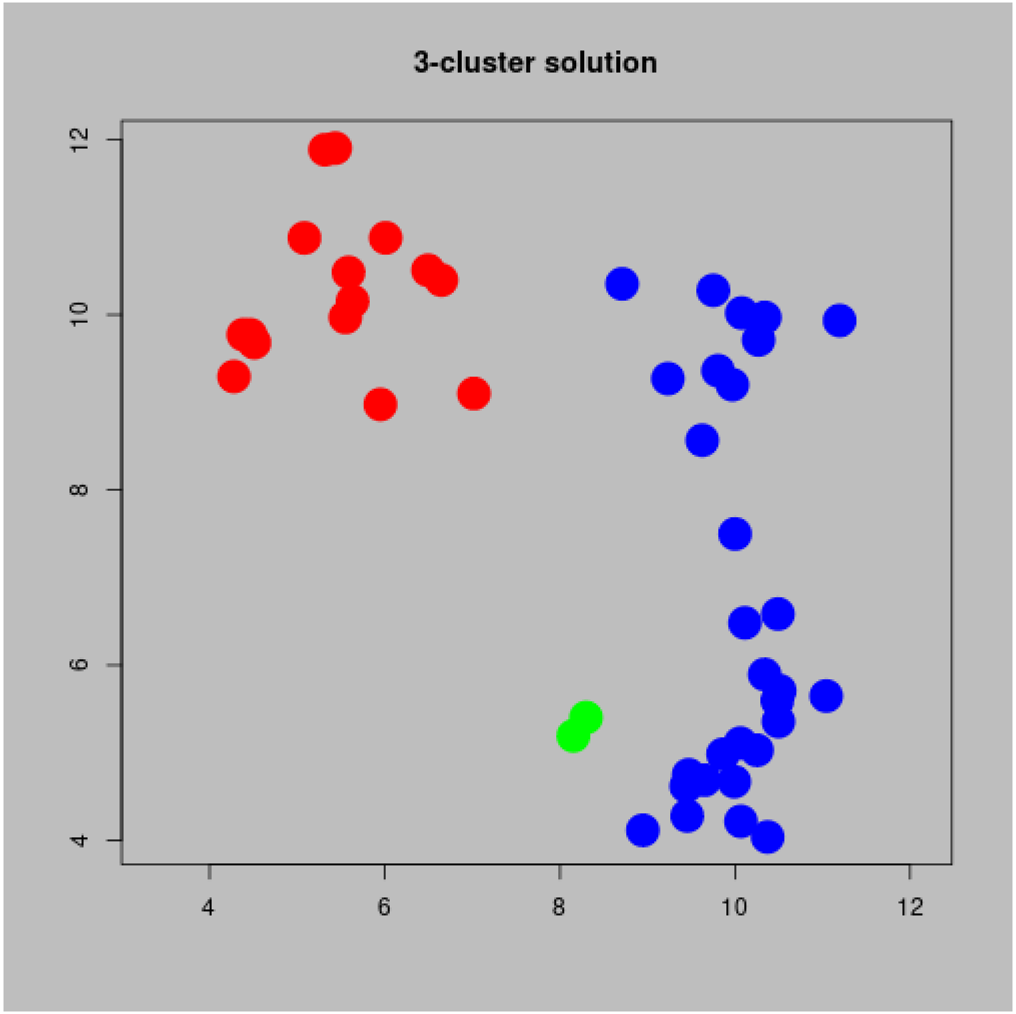
Single Linkage



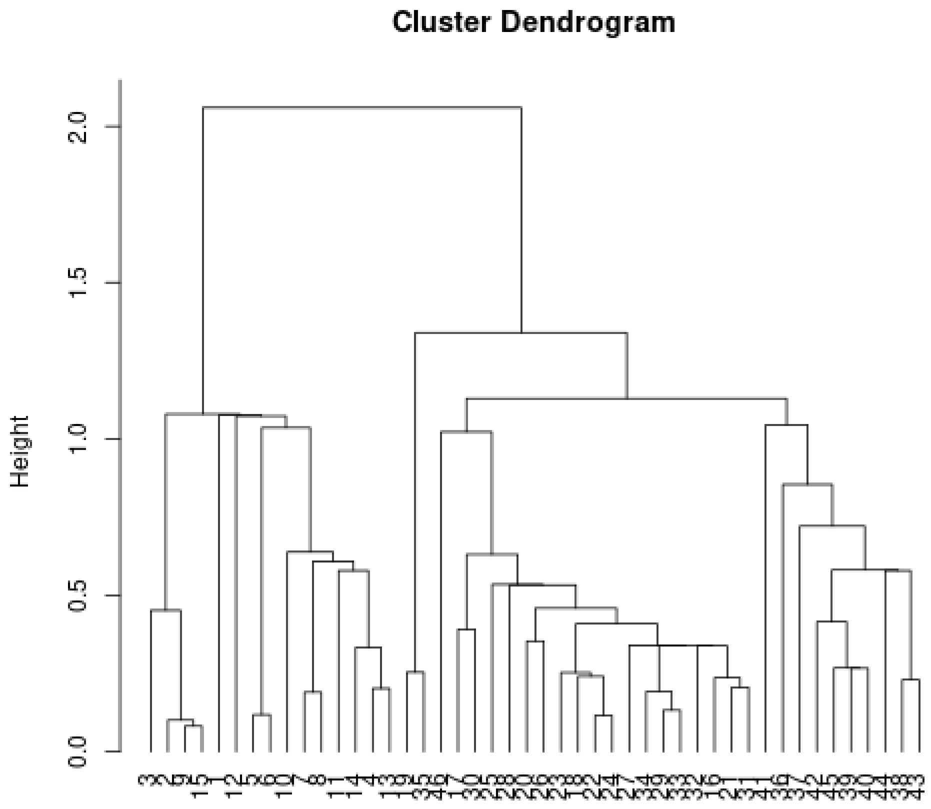
Add one observation



New solution



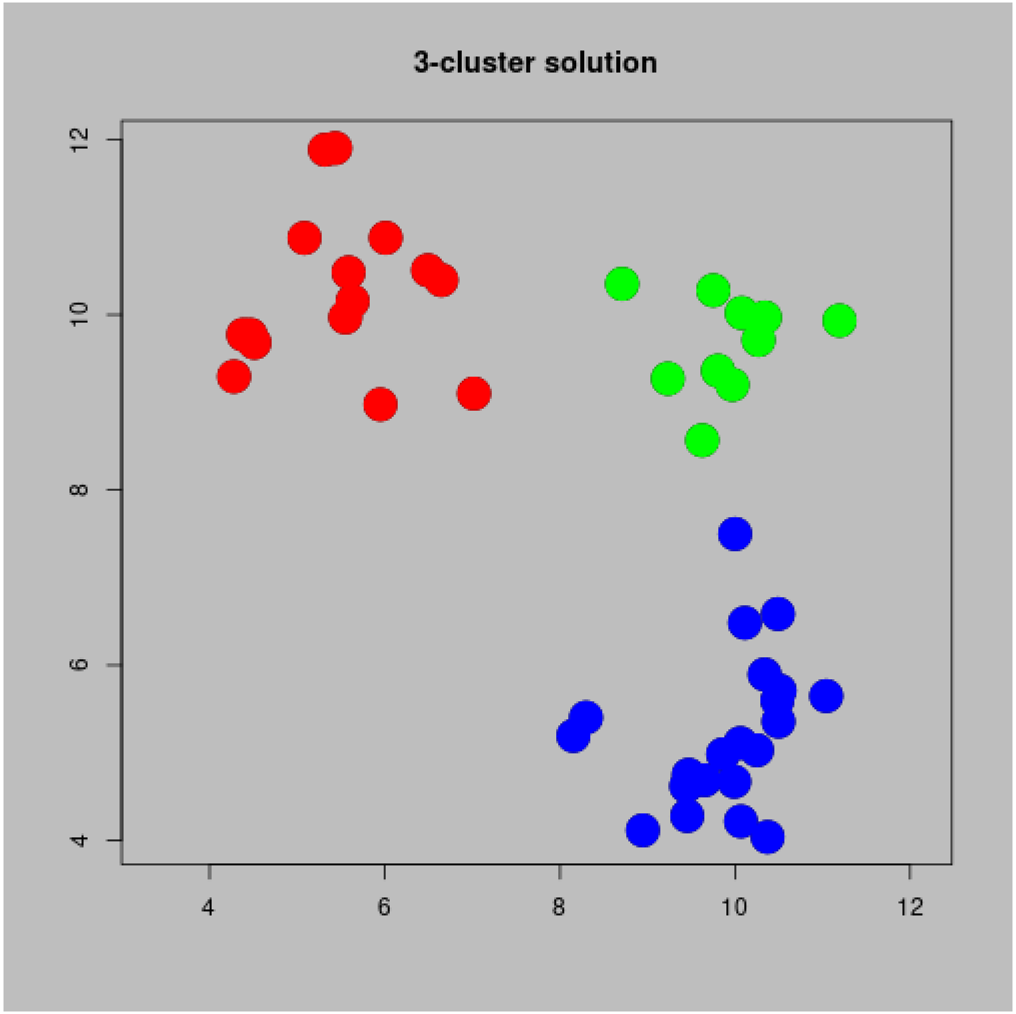
Dendrogram with Chaining



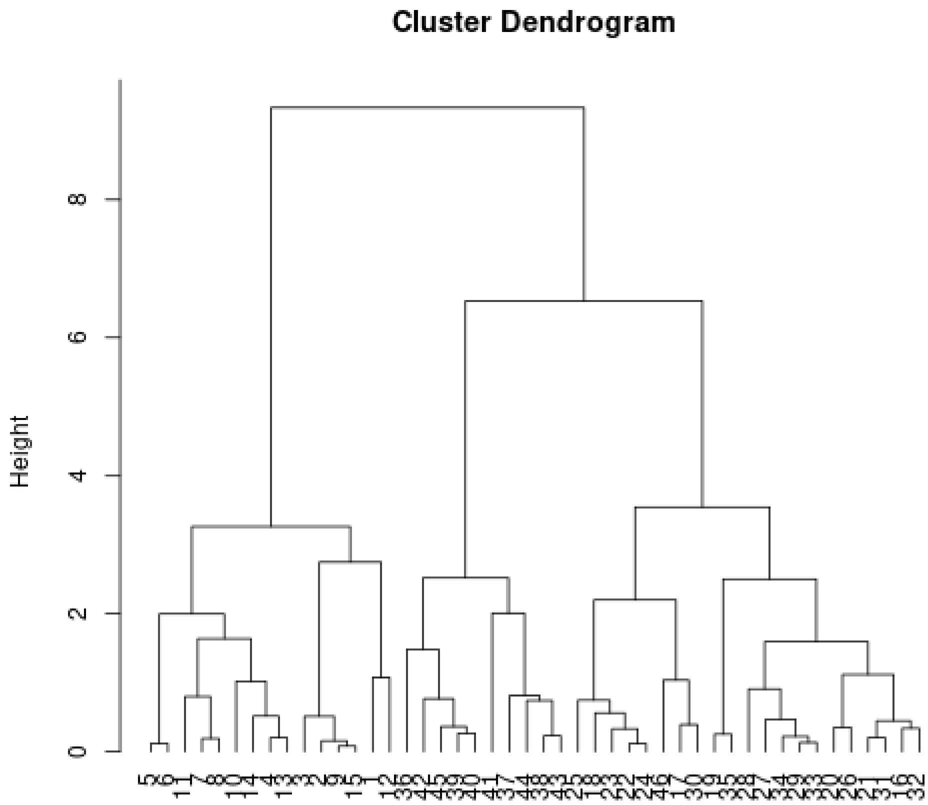
Robustness

- In general adding a single observation should not dramatically change the analysis.
- In this instance the new observation was not even an *outlier*.
- A term used for such an observation is an *inlier*.
- Methods that are not affected by single observations are often called **robust**.
- Let's see if complete linkage is *robust* to the inlier.

Complete Linkage



Complete Linkage: Dendrogram



Disadvantages of CL

- Complete Linkage overcomes *chaining* and is robust to inliers
- However, since the distance between clusters only depends on two observations it can still be sensitive to outliers.
- The following methods are more robust and should be preferred
  - Average Linkage
  - Centroid Method
  - Ward’s Method

Average Linkage

The distance between two clusters can be defined so that it is based on all the pairwise distances between the elements of each cluster.

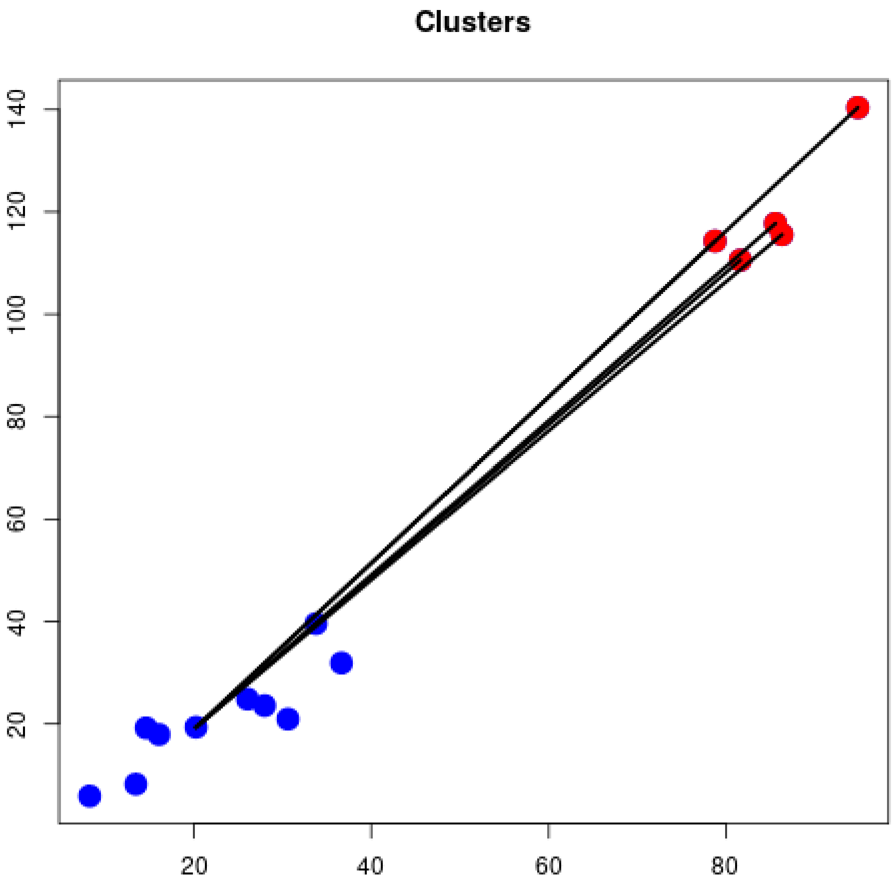
$$D(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{A}||\mathcal{B}|} \sum_{i=1}^{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{B}|} D(\mathbf{a}_i, \mathbf{b}_j)$$

Here $|\mathcal{A}|$ is the number of observations in cluster $\mathcal{A}$ and $|\mathcal{B}|$ is the number of observations in cluster $\mathcal{B}$

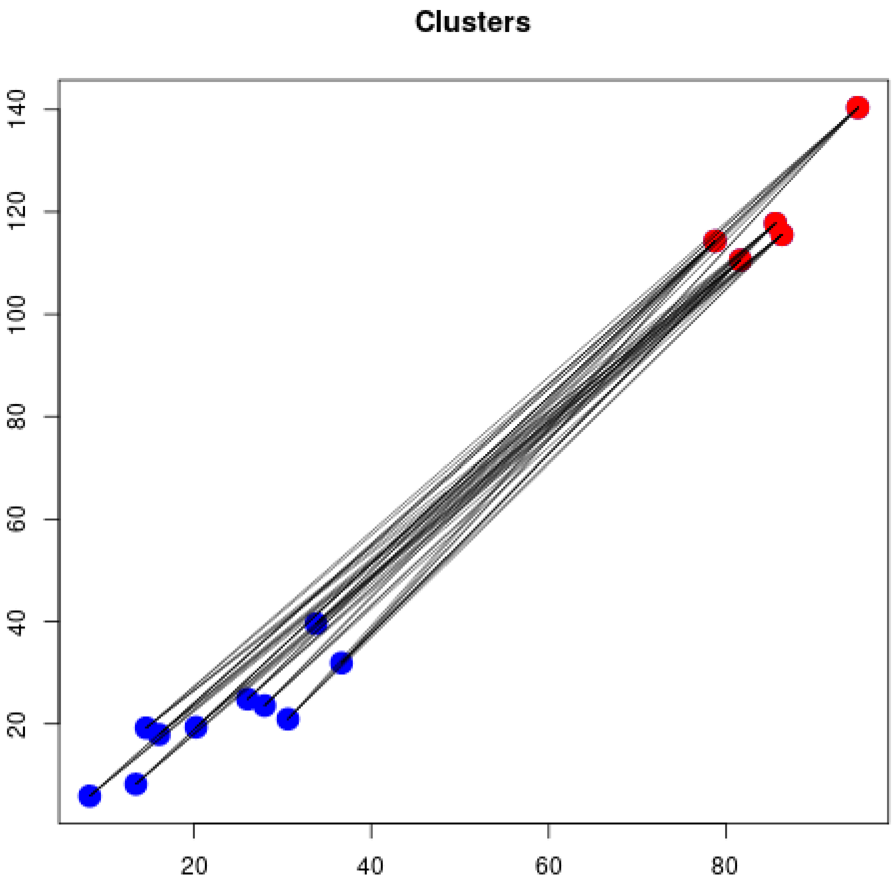
Average Linkage

- Average linkage can be called different things
  - Between groups method.
  - Unweighted Pair Group Method with Arithmetic mean (UPGMA)

Pairwise distances (one obs.)



All pairwise distances



Centroid Method

- The centroid of a cluster can be defined as the mean of all the points in the cluster.
- If $\mathcal{A}$ is a cluster containing the observations $\mathbf{a}$ then the **centroid** of $\mathcal{A}$ is given by.

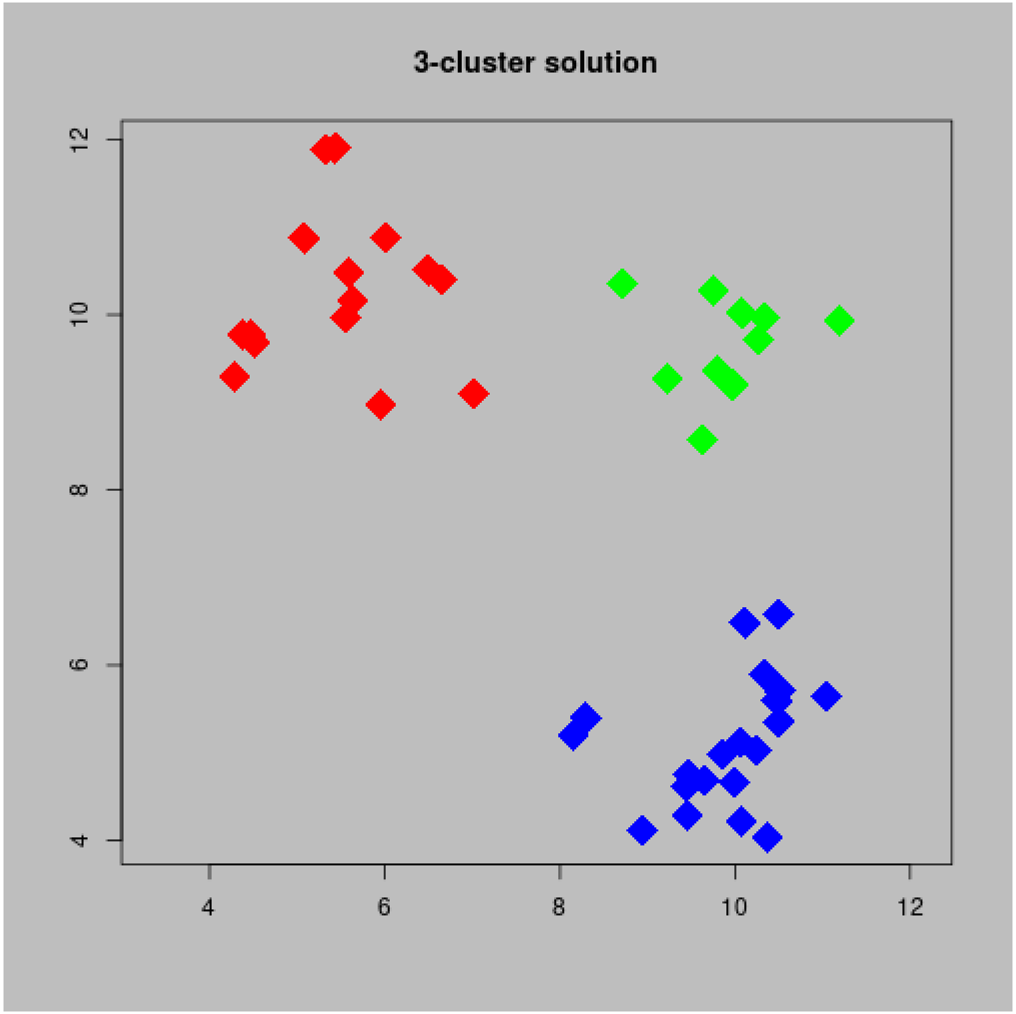
$$\bar{\mathbf{a}} = \frac{1}{|\mathcal{A}|} \sum_{\mathbf{a}_i \in \mathcal{A}} \mathbf{a}_i$$

- The distance between two clusters can then be defined as the distance between the respective centroids.

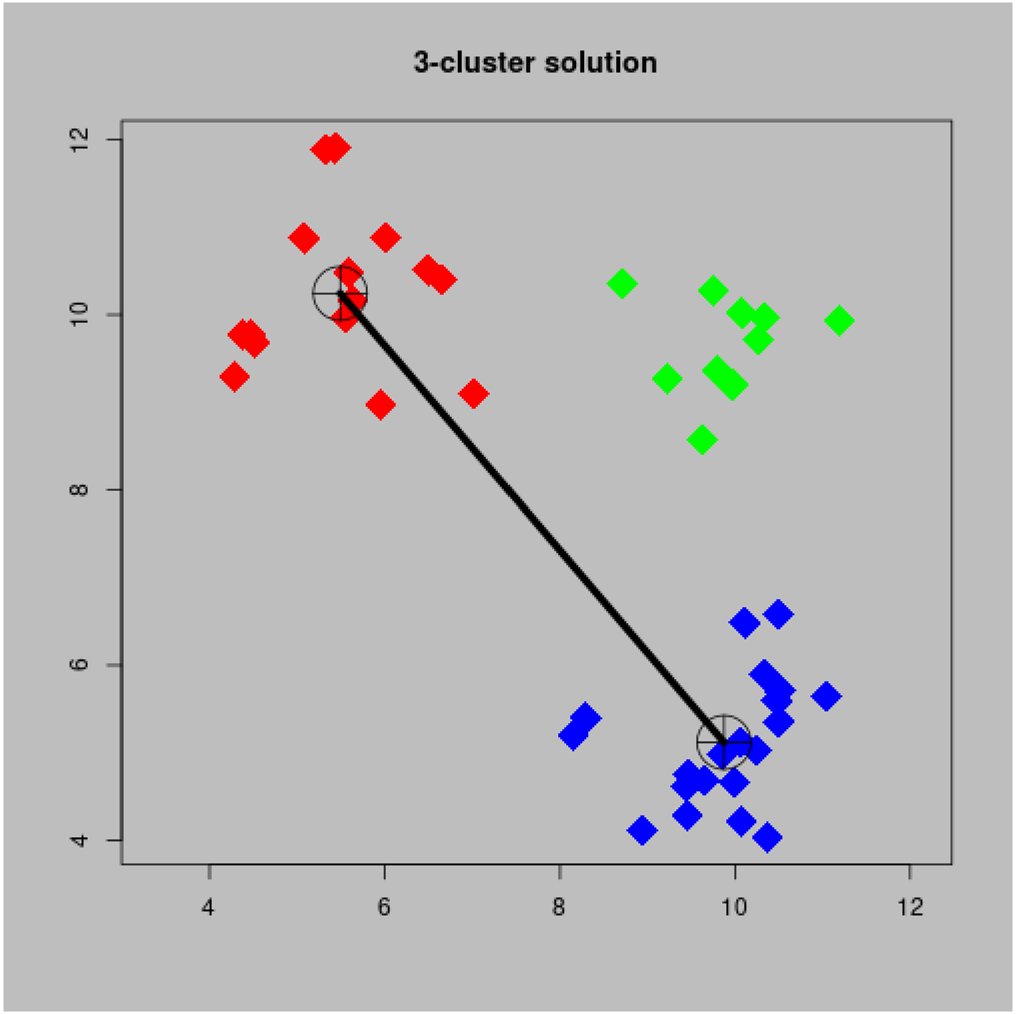
Vector mean

- Recall that $\mathbf{a}_i$ is a vector of attributes, e.g income and age.
- In this case $\bar{\mathbf{a}}$ is also a vector of attributes.
- Each element of $\bar{\mathbf{a}}$ is the mean of a different attribute, e.g. mean income, mean age.

Centroid method



Centroid method



Average Linkage v Centroid

- Consider an example with one variable (although everything works with vectors too).
- Suppose we have the clusters $\mathcal{A} = \{0, 2\}$ and $\mathcal{B} = \{3, 5\}$
- Find the distance $\mathcal{A}$ and $\mathcal{B}$ using
  - Average Linkage
  - Centroid Method

Average Linkage

- Must find distances between all pairs of observations
  - $D(a_1, b_1) = 3$
  - $D(a_1, b_2) = 5$
  - $D(a_2, b_1) = 1$
  - $D(a_2, b_2) = 3$
- Averaging these, the distance is 3.

Centroid method

- First find centroids
  - $\bar{a} = 1$
  - $\bar{b} = 4$
- The distance is 3.
- Here both methods give the same answer but when vectors are used instead they do not give the same answer in general.

Average Linkage v Centroid

- In average linkage
  - Compute the distances between pairs of observations
  - Average these distances
- In the centroid method
  - Average the observations to obtain the centroid of each cluster.
  - Find the distance between centroids

Ward's method

- All methods so far, merge two clusters when the distance between them is small.
- Ward’s method merges two clusters to minimise within cluster variance.

Within Cluster Variance

- The within-cluster variance for a cluster $\mathcal{A}$ is defined as

$$\mathrm{V}_{\mathbf{W}}(\mathcal{A}) = \frac{1}{|\mathcal{A}| - 1} S(\mathcal{A})$$

where

$$S(\mathcal{A}) = \sum_{\mathbf{a}_i \in \mathcal{A}} \left[(\mathbf{a}_i - \bar{\mathbf{a}})' (\mathbf{a}_i - \bar{\mathbf{a}}) \right]$$

Vector notation

- The term $S(\mathcal{A}) = \sum_{\mathbf{a}_i \in \mathcal{A}} (\mathbf{a}_i - \bar{\mathbf{a}})' (\mathbf{a}_i - \bar{\mathbf{a}})$ uses vector notation, but the idea is simple.
- Take the difference of each attribute from its mean (e.g. income, age, etc.)
- Then square them and add together over attributes **and** observations.
- The within cluster variance is a total variance across all attributes.

Ward's algorithm

- At each step we must merge two clusters to form a single cluster.
- Suppose we pick a cluster $\mathcal{A}$ and $\mathcal{B}$ to form a new cluster $\mathcal{C}$.
- Ward's algorithm chooses $\mathcal{A}$ and $\mathcal{B}$ so that $V_W(\mathcal{C})$ is as small as possible.

Wrap-up

Conclusions

- We have covered hierarchical clustering
- In BUSS6002 you will also cover *k-means clustering*.
- An advantage of hierarchical clustering is visualisation via the dendrogram.
- However the ideas of understanding when observations are similar, is useful in many other areas of business analytics.

Questions