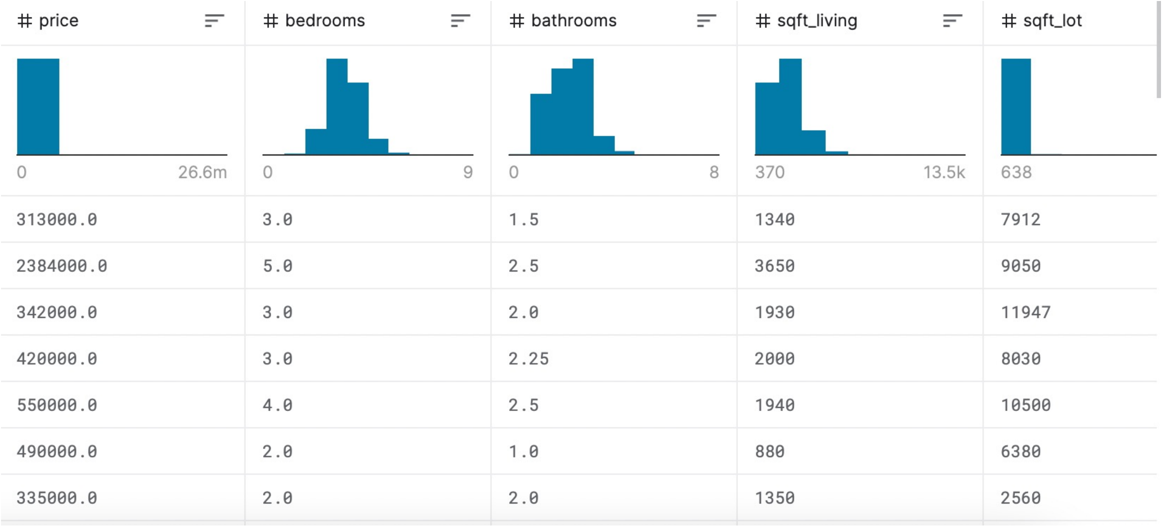
Unsupervised Learning
Analysis Methods for Data Exploration

Data Analysis Steps

- Step 0: Get to know your dataset
  - Visualization (histogram, scatter plot, box plot, barplot, pie chart)
  - Numerical summary (measure of center/spread)
  - Pre-processing (aka cleaning, dplyr package)
- Step 1: Determine your goal
  - Build a dashboard? (Shiny package in R)
  - **Make prediction? Aka Modeling!**
  - **Analyze relationships?**

Unsupervised learning vs Supervised learning



# price	# bedrooms	# bathrooms	# sqft_living	# sqft_lot
313000.0	3.0	1.5	1340	7912
2384000.0	5.0	2.5	3650	9050
342000.0	3.0	2.0	1930	11947
420000.0	3.0	2.25	2000	8030
550000.0	4.0	2.5	1940	10500
490000.0	2.0	1.0	880	6380
335000.0	2.0	2.0	1350	2560

Supervised learning: what features (aka variables) impact the price of a house? What sort of relationship do they have?

Unsupervised learning: find subgroups (aka clusters) in the dataset.

- houses should look *similar* in terms of ALL variables in the same group, AND
- houses should look *different* across other groups.

Unsupervised learning vs Supervised learning

Supervised learning is a collection of models (algorithms) that uses a set of variables to predict a variable of interest.

$$f(x) = y$$

Note: There is always one response variable in any dataset, but there could be multiple predictors.

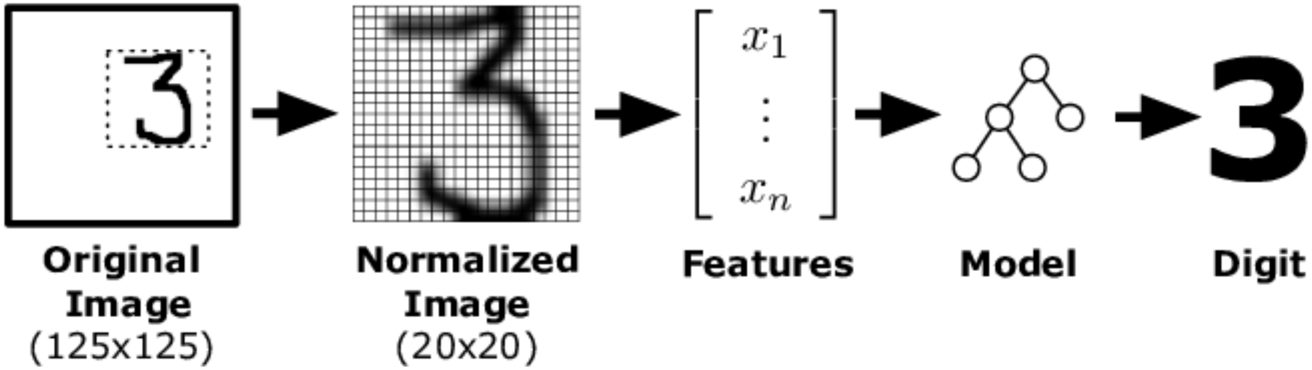
Unsupervised learning vs Supervised learning

Unsupervised learning is a collection of models (algorithms) that find patterns in the dataset.

Note: It does NOT have a response variable, but can have MULTIPLE independent variables.

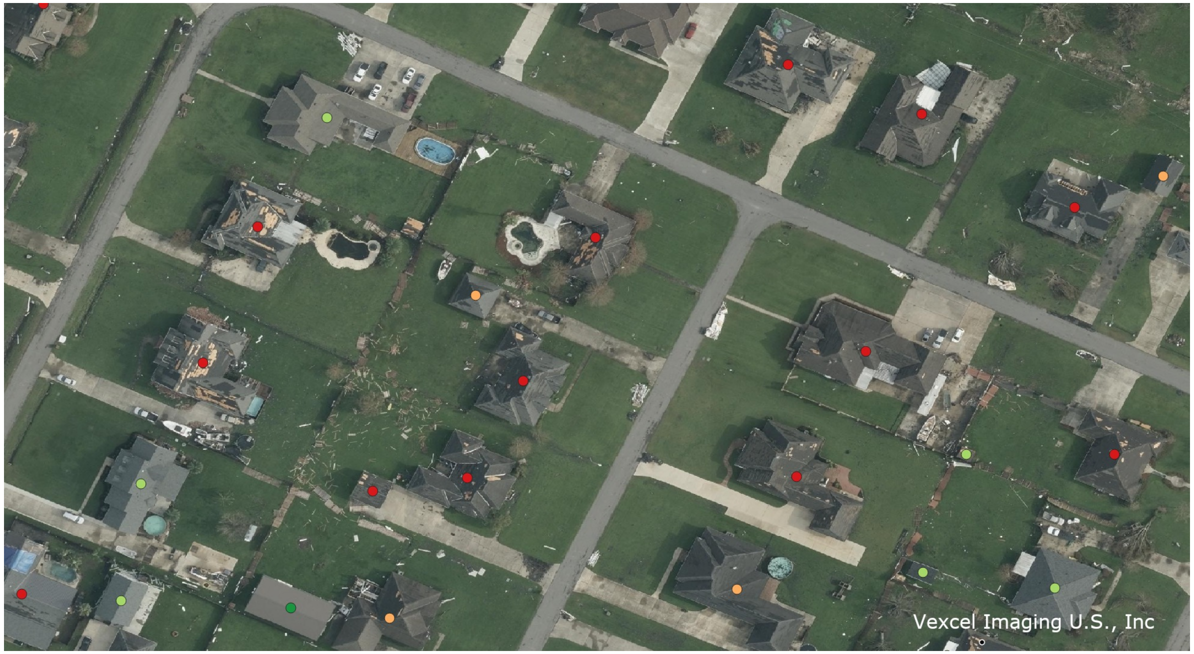
To test your understanding..

1. A digital image of a license plate is obtained from a camera mounted on a toll booth. For each character in the plate ID, we would like to assign the correctly identified digit from the set (A, B, C, …, Z, 0, 1, …, 9}.



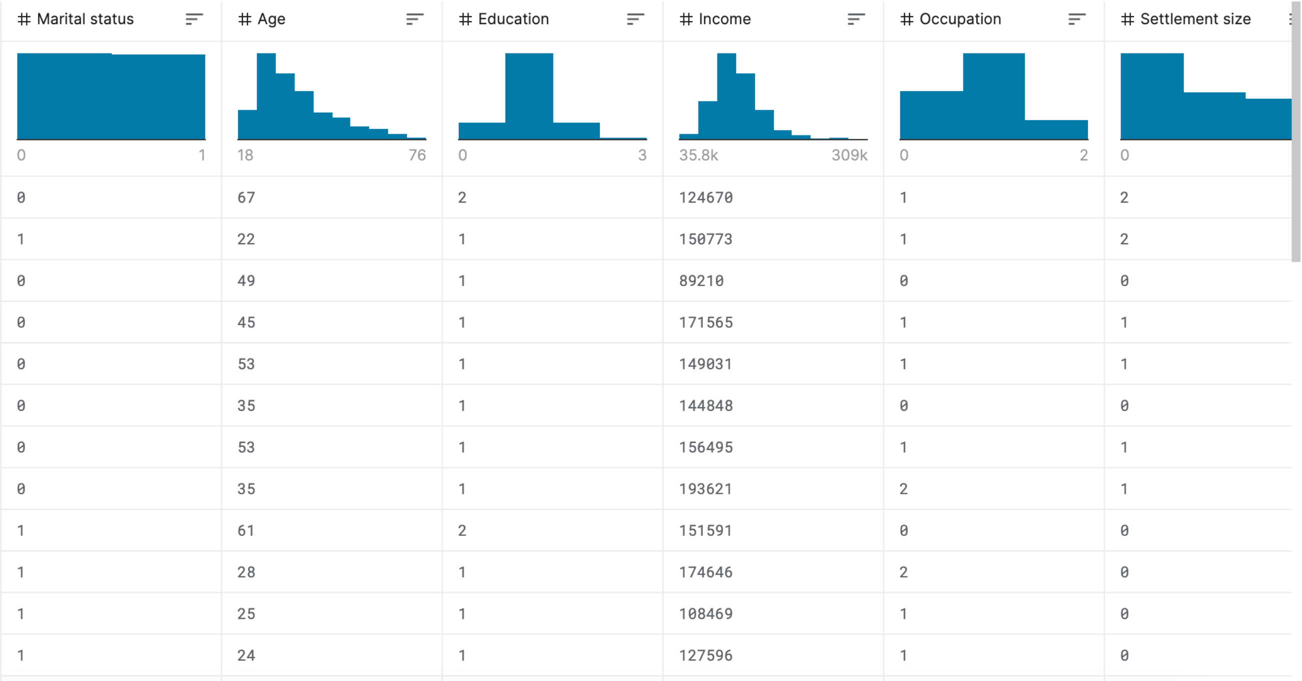
To test your understanding..

2. Given a 640x640 pixel grid on a satellite image around each house, estimate hurricane damage by assessing roof damage: Dark Green = Excellent; Light Green = Good; Yellow = Fair; Orange = Poor; Red = Severe



To test your understanding..

3. Given the following information about customers who purchase items from a grocery store, find out certain patterns/similarities in customers’ characteristics.



# Marital status	# Age	# Education	# Income	# Occupation	# Settlement size
0	67	2	124670	1	2
1	22	1	150773	1	2
0	49	1	89210	0	0
0	45	1	171565	1	1
0	53	1	149031	1	1
0	35	1	144848	0	0
0	53	1	156495	1	1
0	35	1	193621	2	1
1	61	2	151591	0	0
1	28	1	174646	2	0
1	25	1	108469	1	0
1	24	1	127596	1	0

The models

Supervised learning models:

- Linear regression
- K-nearest neighborhood (KNN)
- Support Vector Machines
- Naïve Bayes
- Decision Trees (and their extensions)
- Neural Network
- …

Unsupervised learning models:

- Clustering
- Principal Component Analysis
- Association rules
- …

How to measure dissimilarity

Distance measures: a larger “distance” in between two sample units means they are less similar, a smaller distance means they are more similar.

Common measures:

- Euclidean distance
- Manhattan distance (aka city block distance)
- Minkowski distance (in between of the first two)

Look at these two grocery stores..

	Item 1	Item 2
Store 1	$4	$3
Store 2	$4.5	$3.5

How similar are these two stores in terms of pricing?

Look at these two grocery stores..

	Item 1	Item 2
Store 1	$4	$3
Store 2	$4.5	$3.5

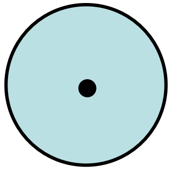
How similar are these two stores in terms of pricing?

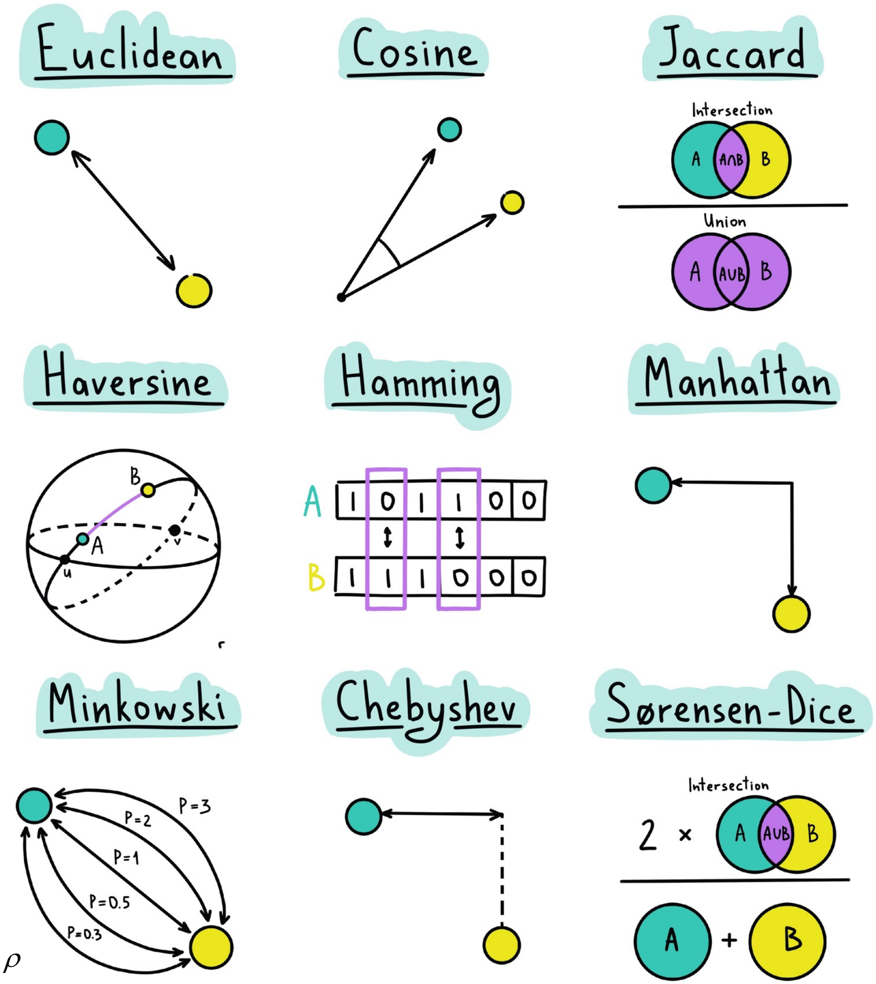
Using Euclidean distance:
square root of [(4-4.5)^2 + (3-3.5)^2]

Using Manhattan distance:
absolute value for (4 – 4.5) + absolute value for (3 – 3.5)

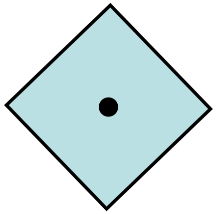
Summary of Distance Measures for Continuous Data

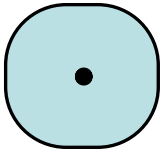
$$d_{ij} = \left(\sum_{k=1}^{p} (x_{ik} - x_{jk})^2 \right)^{1/2}$$





$$d_{ij} = \sum_{k=1}^{p} | x_{ik} - x_{jk} |$$





$$d_{ij} = \left(\sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right|^{\rho} \right)^{1/\rho} \quad \rho \geq 1$$

ABACUS.AI

Question

Suppose we are given the following 3x4 data matrix

$$\mathbf{X} = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 1 & 4 & 0 & 2 \\ 3 & 0 & 1 & 1 \end{bmatrix}$$

Use the Euclidian metric to produce a distance matrix **D** (leave your entries as square roots). Hint: How many things do you need to compute?

Matrices

Definition: An m x n **matrix** is a rectangular array of elements (e.g. real numbers), arranged in m rows and n columns. The elements of the matrix are called the **entries**. The expression m x n denotes the **size** of the matrix.

Examples:

$$\mathbf{A} = \begin{bmatrix} 1 & 6 & -2 & 10 \\ 6 & 3 & 3 & 4 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -2 & 23 \\ 0.5 & 7 \\ 1 & 2.76 \end{bmatrix}$$

A is a 2 x 4 matrix

B is a 3 x 2 matrix

If **A** is an m x n matrix, we will denote the entry in the i^{th} row and the j^{th} column by a_{ij}.

Examples:

$$\mathbf{A} = \begin{bmatrix} 1 & 6 & -2 & 10 \\ 6 & 3 & 3 & 4 \end{bmatrix}$$

$a_{23} = 3$

$a_{14} = 10$

$$\mathbf{B} = \begin{bmatrix} -2 & 23 \\ 0.5 & 7 \\ 1 & 2.76 \end{bmatrix}$$

$b_{11} = -2$

$b_{22} = 7$

More Facts About Matrices

- A **square matrix** is an $n \times n$ matrix, i.e., a square matrix has the same number of rows and columns.

  Examples:

  $$\mathbf{D} = \begin{bmatrix} -2 & 23 & 4 \\ 0.5 & 7 & 4 \\ 1 & 2.76 & 4 \end{bmatrix}$$

  $$\mathbf{B} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix}$$

  $$\mathbf{A} = [21]$$

- A matrix **A** is **symmetric** if it is square and if $a_{ij} = a_{ji}$ for all $1 \leq i, j \leq n$.

  Examples:

  $$\mathbf{A} = \begin{bmatrix} -2 & 0.5 & 4 \\ 0.5 & 7 & 18 \\ 4 & 18 & 4 \end{bmatrix}$$

  $$\mathbf{B} = \begin{bmatrix} 1 & 2 \\ 2 & -3 \end{bmatrix}$$

Data Matrix

Crime data as above, scaled to rate per 100,000 population.

		Pop	Violent crime	Murder	Rob-bery	Aggra-vated assault	Property crime	Burglary	Larceny-theft	Motor vehicle theft	Arson
Buffalo	2007	273,832	1,275	20	560	635	5,893	1,603	3,461	829	43
New York	2007	8,220,196	614	6	265	332	1,819	254	1,403	161	68
Rochester	2007	206,686	1,133	24	497	552	5,419	1,238	3,388	794	98
Syracuse	2007	139,880	1,026	14	319	646	4,264	1,276	2,587	401	34
Yonkers	2007	198,071	443	5	214	202	1,521	324	1,007	190	18

This information is condensed into the following data matrix.

$$\begin{bmatrix} 1{,}275 & 20 & 560 & 635 & 5{,}893 & 1{,}603 & 3{,}461 & 829 & 43 \\ 614 & 6 & 265 & 332 & 1{,}819 & 254 & 1{,}403 & 161 & 68 \\ 1{,}133 & 24 & 497 & 552 & 5{,}419 & 1{,}238 & 3{,}388 & 794 & 98 \\ 1{,}026 & 14 & 319 & 646 & 4{,}264 & 1{,}276 & 2{,}587 & 401 & 34 \\ 443 & 5 & 214 & 202 & 1{,}521 & 324 & 1{,}007 & 190 & 18 \end{bmatrix}$$

So how do we use this information to form clusters??

Data and Dissimilarity Matrices

We begin with an n x p multivariate data matrix containing n objects as rows that are to be grouped based on p attributes arranged as columns

Data Matrix:

Traits →

Objects ↓

***i* th object** →

***j* th attribute** →

$$\begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{bmatrix}$$

From this we construct a **Dissimilarity Matrix** whose (i, j) entry is $d(O_i, O_j)$.

Dissimilarity Matrix:

$$\begin{bmatrix} d(O_1,O_1) & d(O_1,O_2) & \cdots & & d(O_1,O_n) \\ d(O_2,O_1) & d(O_2,O_2) & \cdots & & d(O_2,O_n) \\ d(O_3,O_1) & d(O_3,O_2) & & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ d(O_n,O_1) & d(O_n,O_2) & \cdots & \cdots & d(O_n,O_n) \end{bmatrix}$$

Dissimilarity between the 3rd and 2nd object →

Distance Matrix

What must be true regarding a dissimilarity matrix?

1. **It is square;** if there are n objects, then D is $n \times n$.

2. **It has 0s on its main diagonal.** That is, $d_{ii} = d(O_i, O_i) = 0$ for $1 \leq i \leq n$.

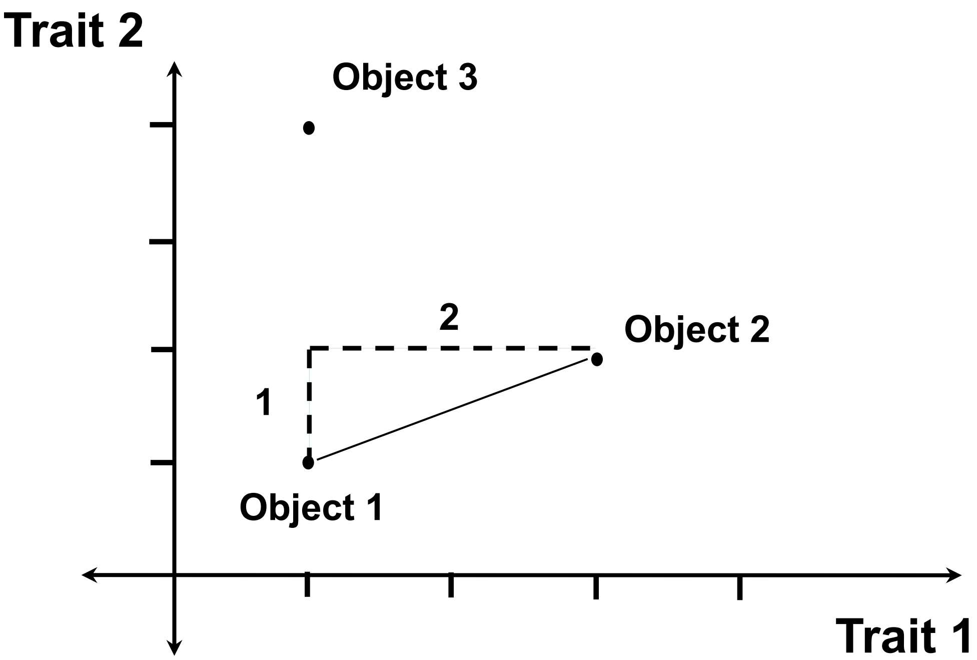
3. **It is symmetric** since $d_{ij} = d(O_i, O_j) = d(O_j, O_i) = d_{ji}$ for $1 \leq i, j \leq n$.

$$\mathbf{D} = \begin{bmatrix} 0 & d(O_1,O_2) & \cdots & & d(O_1,O_n) \\ d(O_2,O_1) & 0 & \cdots & & d(O_2,O_n) \\ d(O_3,O_1) & d(O_3,O_2) & & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ d(O_n,O_1) & d(O_n,O_2) & \cdots & \cdots & 0 \end{bmatrix}$$

Example of a Distance Measure

As an example, suppose we are measuring two continuous traits on our objects with the same scale. Furthermore, suppose our data matrix is given as

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 3 & 2 \\ 1 & 4 \end{bmatrix}$$



Plot the objects on the graph to the left.

What is the Euclidean (physical) distance between O_1 and O_2?

$$d(O_1, O_2) = \sqrt{1^2 + 2^2} = \sqrt{5}$$

Back to the question from earlier

Suppose we are given the following 3x4 data matrix

$$\mathbf{X} = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 1 & 4 & 0 & 2 \\ 3 & 0 & 1 & 1 \end{bmatrix}$$

Use the Euclidian metric to produce a distance matrix **D** (leave your entries as square roots). Hint: How many things do you need to compute?

$$d_{12} = \left(\sum_{k=1}^{4} (x_{1k} - x_{2k})^2 \right)^{1/2} = \sqrt{(0-1)^2 + (2-4)^2 + (1-0)^2 + (3-2)^2} = \sqrt{7}$$

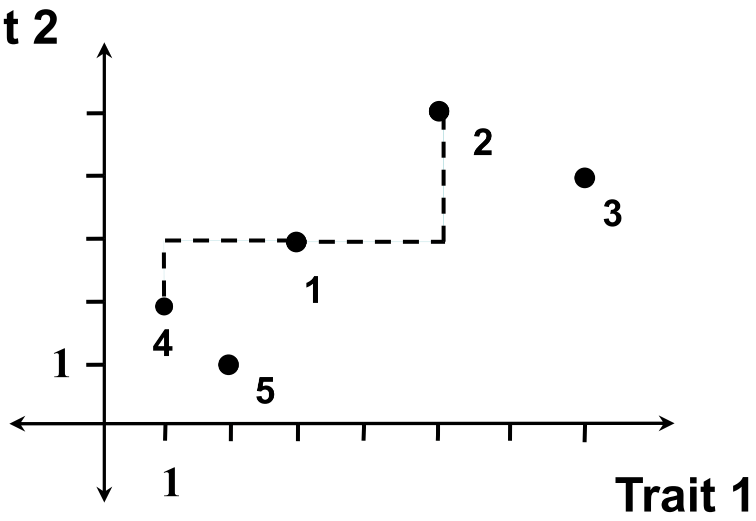
$$d_{13} = \left(\sum_{k=1}^{4} (x_{1k} - x_{3k})^2 \right)^{1/2} = \sqrt{(0-3)^2 + (2-0)^2 + (1-1)^2 + (3-1)^2} = \sqrt{17}$$

$$d_{23} = \left(\sum_{k=1}^{4} (x_{2k} - x_{3k})^2 \right)^{1/2} = \sqrt{22}$$

$$\mathbf{D} = \begin{bmatrix} 0 & \sqrt{7} & \sqrt{17} \\ \sqrt{7} & 0 & \sqrt{22} \\ \sqrt{17} & \sqrt{22} & 0 \end{bmatrix}$$

Another question

Consider the following five objects, numbered 1 thru 5, that are to be clustered using two continuous variables. Using the Manhattan metric, find the distance matrix, **D**, for the objects.



$$\mathbf{D} = \begin{bmatrix} 0 & 4 & 5 & 3 & 3 \\ 4 & 0 & 3 & 7 & 7 \\ 5 & 3 & 0 & 8 & 8 \\ 3 & 7 & 8 & 0 & 2 \\ 3 & 7 & 8 & 2 & 0 \end{bmatrix}$$

Which two objects are “closest” ?

Computing Distances in R

Starting with an *n* x *p* data matrix (or data frame), we can compute the distance matrices in R using the `dist` function:

```
> X<-matrix(c(1,1,2,3,1,4),nrow=3,byrow=T)
> X
     [,1] [,2]
[1,]    1    1
[2,]    2    3
[3,]    1    4
> dist(X)
         1        2
2 2.236068
3 3.000000 1.414214
```

The default method is “euclidean” for the Euclidian distance. Several other methods are available.

```
> dist(X,method="manhattan")
  1 2
2 3
3 3 2
```

```
> dist(X,method="minkowski",p=3,upper=T)
         1        2        3
1          2.080084 3.000000
2 2.080084          1.259921
3 3.000000 1.259921
```

Only the lower portion of the distance matrix is printed unless specified with `upper=T`. You can specify to have the diagonal printed with `diag=T`.

Computing Distances in R

The function `dist` returns an object of class "dist". This is a data structure that efficiently stores the distance data because of the redundancy present in a regular distance matrix.

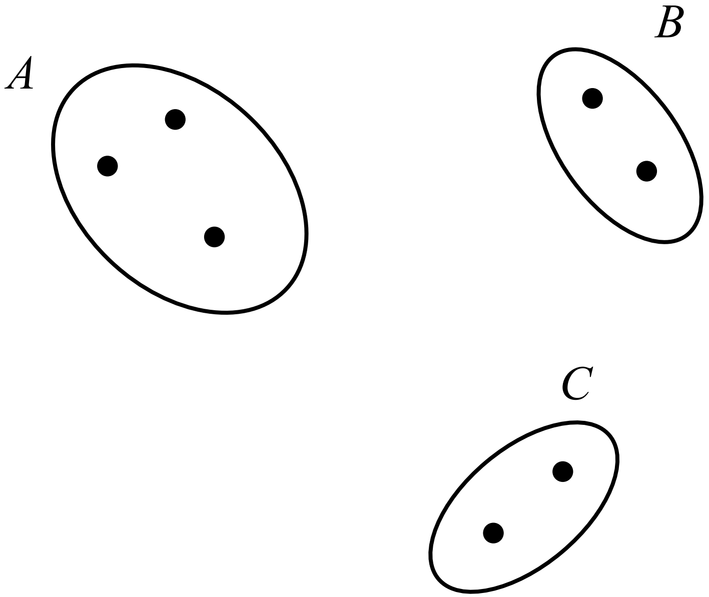
A distance matrix can be obtained from this data structure using the `as.matrix` function.

```
> Dist_X<-dist(X)
> Dist_X
         1        2
2 2.236068
3 3.000000 1.414214
> Dist_X_mat<-as.matrix(Dist_X)
> Dist_X_mat
         1        2        3
1 0.000000 2.236068 3.000000
2 2.236068 0.000000 1.414214
3 3.000000 1.414214 0.000000
```

Most downstream analysis, however, expects a `dist` object as input and so this information is usually kept in this form.

Inter-group Proximity Measures

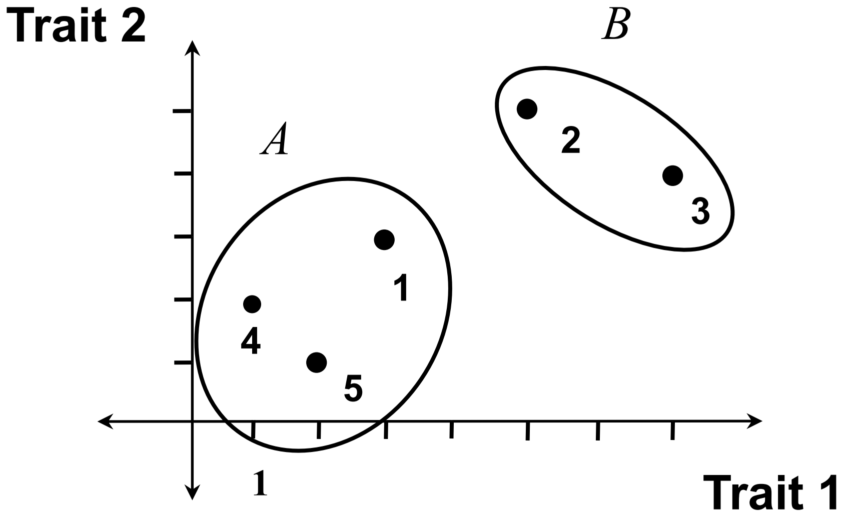
To this point, we have concerned ourselves with production a distance matrix, **D**, from an $n \times p$ data matrix, **X**. It will also become necessary to determine the distance between two *groups* of objects.



What should we use for a measurement of the distance between the two groups *A* and *B*?

Example: Nearest-Neighbor Distance

Consider the following five objects in the Euclidian plane (two continuous traits) with the Manhattan metric.



Distance Matrix

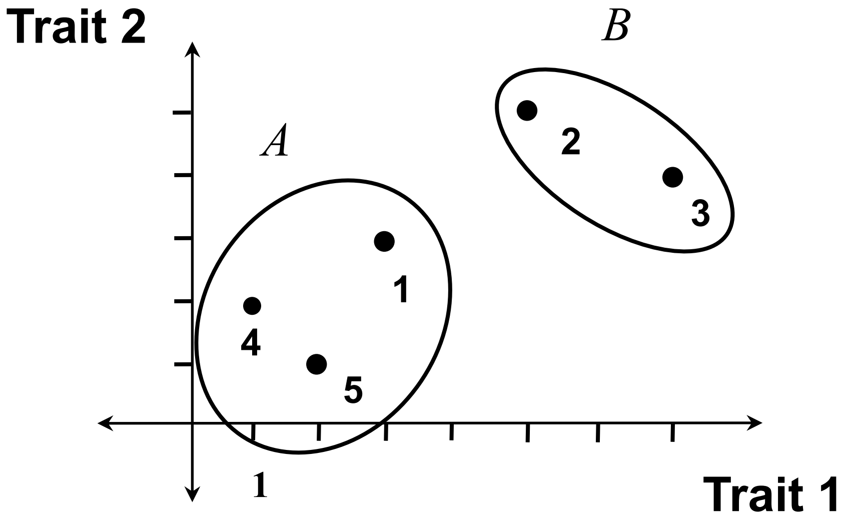
$$\mathbf{D} = \begin{bmatrix} 0 & 4 & 5 & 3 & 3 \\ 4 & 0 & 3 & 7 & 7 \\ 5 & 3 & 0 & 8 & 8 \\ 3 & 7 & 8 & 0 & 2 \\ 3 & 7 & 8 & 2 & 0 \end{bmatrix}$$

What is the inter group distance between groups *A* and *B* using the nearest-neighbor distance?

$$\begin{aligned} d^{\mathrm{S}}(A, B) &= \min\{d(a,b) : a \in A, b \in B\} \\ &= \min\{d_{12}, d_{13}, d_{42}, d_{43}, d_{52}, d_{53}\} \\ &= \min\{4, 5, 7, 8, 7, 8\} = 4 \end{aligned}$$

Example: Furthest-Neighbor Distance

Again, consider the following five objects in the Euclidian plane (two continuous traits) with the Manhattan metric.



Distance Matrix

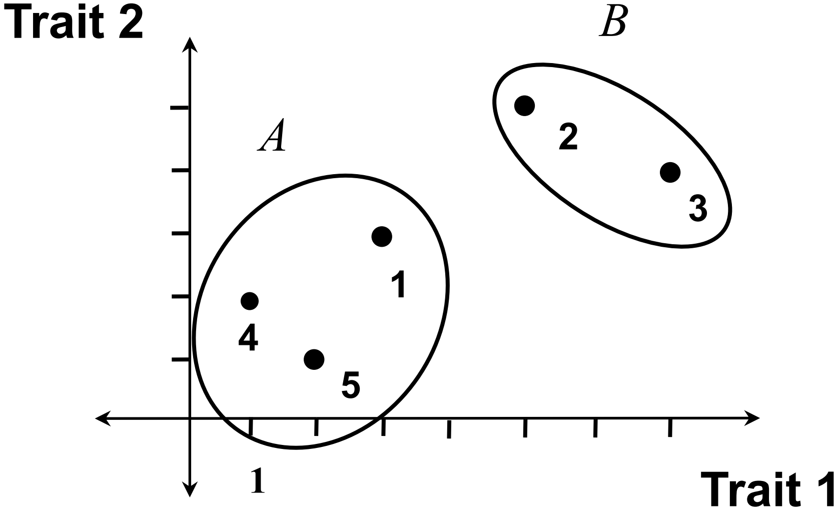
$$\mathbf{D} = \begin{bmatrix} 0 & 4 & 5 & 3 & 3 \\ 4 & 0 & 3 & 7 & 7 \\ 5 & 3 & 0 & 8 & 8 \\ 3 & 7 & 8 & 0 & 2 \\ 3 & 7 & 8 & 2 & 0 \end{bmatrix}$$

What is the inter group distance between groups *A* and *B* using the furthest-neighbor distance?

$$\begin{aligned} d^{\mathrm{C}}(A, B) &= \max\{d(a,b) : a \in A, b \in B\} \\ &= \max\{d_{12}, d_{13}, d_{42}, d_{43}, d_{52}, d_{53}\} \\ &= \max\{4, 5, 7, 8, 7, 8\} = 8 \end{aligned}$$

Example: Average-Neighbor Distance

Again, consider the following five objects in the Euclidian plane (two continuous traits) with the Manhattan metric.



Distance Matrix

$$\mathbf{D} = \begin{bmatrix} 0 & 4 & 5 & 3 & 3 \\ 4 & 0 & 3 & 7 & 7 \\ 5 & 3 & 0 & 8 & 8 \\ 3 & 7 & 8 & 0 & 2 \\ 3 & 7 & 8 & 2 & 0 \end{bmatrix}$$

What is the inter-group distance between groups *A* and *B* using the average-neighbor distance?

$$d^A(A,B) = \frac{1}{|A||B|}\sum_{\substack{a\in A \\ b\in B}} d(a,b) = \frac{1}{3\cdot 2}\left[d_{12}+d_{13}+d_{42}+d_{43}+d_{52}+d_{53}\right]$$

$$= \frac{1}{6}\left[4+5+7+8+7+8\right] = \frac{39}{6} = \frac{13}{2}$$

How to use the distances calculated?: Hierarchical Clustering (aka “bottom-up”)

Suppose we have the following distance matrix for 4 observations

Starting with our dissimilarity matrix, we see that object 1 and object 2 have the smallest distance between them (1.0) thus the first cluster is taken to be {1, 2}.

$$\begin{array}{c|cccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} \\ \hline \mathbf{1} & 0 & 1.0 & 4.2 & 2 \\ \mathbf{2} & 1.0 & 0 & 4.32 & 2.24 \\ \mathbf{3} & 4.2 & 4.32 & 0 & 2.2 \\ \mathbf{4} & 2 & 2.24 & 2.2 & 0 \end{array}$$

The next stage requires the construction of a new dissimilarity matrix that reflects the new state of the system. We now need a method for measuring the distances between the group {1, 2} and the objects 3 and 4.

$$\begin{array}{c|ccc} & \mathbf{\{1,2\}} & \mathbf{3} & \mathbf{4} \\ \hline \mathbf{\{1,2\}} & 0 & ???? & ???? \\ \mathbf{3} & ???? & 0 & 2.2 \\ \mathbf{4} & ???? & 2.2 & 0 \end{array}$$

What method should we use??

Example Using Single Linkage Clustering

Using Single Linkage clustering (nearest-neighbor), the clusters and distance matrices are as follows:

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1.0 & 4.2 & 2 \\ 2 & 1.0 & 0 & 4.32 & 2.24 \\ 3 & 4.2 & 4.32 & 0 & 2.2 \\ 4 & 2 & 2.24 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|ccc} & \{1,2\} & 3 & 4 \\ \hline \{1,2\} & 0 & 4.2 & 2 \\ 3 & 4.2 & 0 & 2.2 \\ 4 & 2 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|cc} & \{1,2,4\} & 3 \\ \hline \{1,2,4\} & 0 & 2.2 \\ 3 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|c} & \{1,2,3,4\} \\ \hline \{1,2,3,4\} & 0 \end{array}$$

The Partition Series

Thus, using Single Linkage clustering, we obtain the following series of partitions of the data.

	Members
Partition 1	**{{1}, {2}, {3}, {4}}**
Partition 2	**{{1, 2}, {3}, {4}}**
Partition 3	**{{1, 2, 4}, {3}}**
Partition 4	**{{1, 2, 4, 3}}**

Example Using Complete Linkage Clustering

Using Complete Linkage clustering (farthest-neighbor), the clusters and distance matrices are as follows:

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1.0 & 4.2 & 2 \\ 2 & 1.0 & 0 & 4.32 & 2.24 \\ 3 & 4.2 & 4.32 & 0 & 2.2 \\ 4 & 2 & 2.24 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|ccc} & \{1,2\} & 3 & 4 \\ \hline \{1,2\} & 0 & 4.32 & 2.24 \\ 3 & 4.32 & 0 & 2.2 \\ 4 & 2.24 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|cc} & \{1,2\} & \{3,4\} \\ \hline \{1,2\} & 0 & 4.32 \\ \{3,4\} & 4.32 & 0 \end{array}$$

$$\begin{array}{c|c} & \{1,2,3,4\} \\ \hline \{1,2,3,4\} & 0 \end{array}$$

The Partition Series

Thus, using Complete Linkage clustering, we obtain the following series of partitions of the data.

	Members
Partition 1	**{{1}, {2}, {3}, {4}}**
Partition 2	**{{1, 2}, {3}, {4}}**
Partition 3	**{{1, 2}, {3,4}}**
Partition 4	**{{1, 2, 4, 3}}**

Example Using Average Linkage Clustering

Using Average Linkage clustering (average-neighbor), the clusters and distance matrices are as follows:

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1.0 & 4.2 & 2 \\ 2 & 1.0 & 0 & 4.32 & 2.24 \\ 3 & 4.2 & 4.32 & 0 & 2.2 \\ 4 & 2 & 2.24 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|ccc} & \{1,2\} & 3 & 4 \\ \hline \{1,2\} & 0 & 4.26 & 2.12 \\ 3 & 4.26 & 0 & 2.2 \\ 4 & 2.12 & 2.2 & 0 \end{array}$$

$$\begin{array}{c|cc} & \{1,2,4\} & 3 \\ \hline \{1,2,4\} & 0 & 3.57 \\ 3 & 3.57 & 0 \end{array}$$

$$\begin{array}{c|c} & \{1,2,3,4\} \\ \hline \{1,2,3,4\} & 0 \end{array}$$

The Partition Series

Thus, using Average Linkage clustering, we obtain the following series of partitions of the data.

	Members
Partition 1	{{1}, {2}, {3}, {4}}
Partition 2	{{1, 2}, {3}, {4}}
Partition 3	{{1, 2, 4}, {3}}
Partition 4	{{1, 2, 4, 3}}

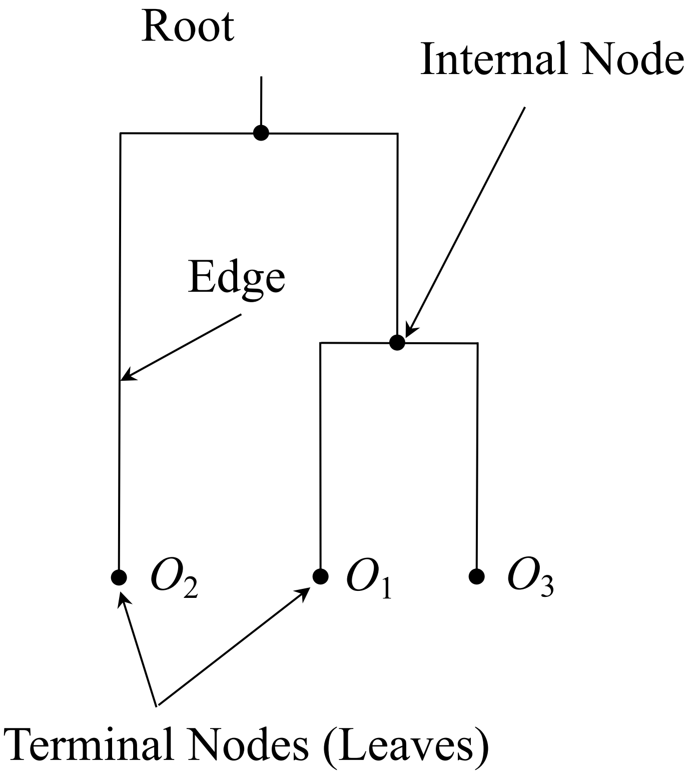
Dendrograms

A **dendrogram** is a graphical representation of a clustering procedure. It is displayed as a **tree diagram** where the clusters are represented by nodes.

Usually, a dendrogram will have two edges emanating (downward) from each non-terminal node making it a **binary tree**. The terminal nodes are also called **leaves**.

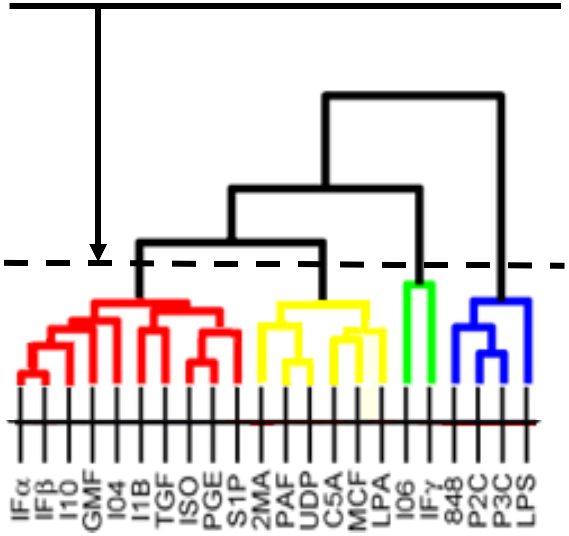
The **topology** of the dendrogram is the structure of the furcations without regard to the distances (or objects that have been clustered).

What information can we infer about the clustering procedure from the dendrogram to the right?



Objects 1 and 3 are the closer than either is to Object 2.

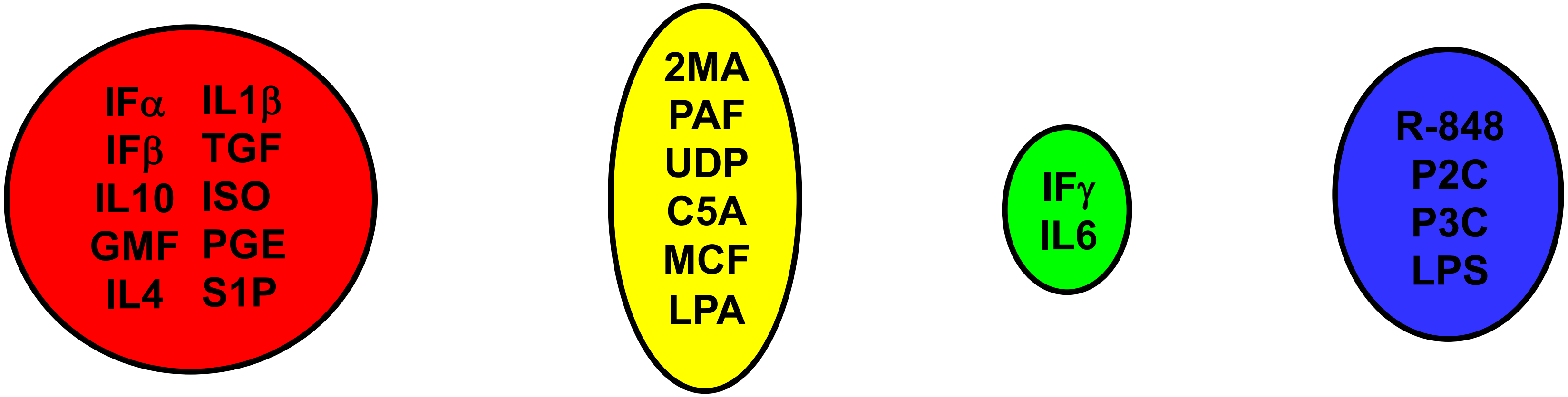
Using a Dendrogram to Identify Clusters



As we move the horizontal line from top to bottom on a weighted dendrogram we break the large group into finer and finer sub-groups.

Moving the line to the dotted line produces four such sub-groups, as it crosses the dendrogram a total of four times.

Portions of the dendrogram may be **colored** to aid in the viewing of the different clusters. In the example above, breaking the object set into four clusters produces the following groups



IFα IL1β
IFβ TGF
IL10 ISO
GMF PGE
IL4 S1P

2MA
PAF
UDP
C5A
MCF
LPA

IFγ
IL6

R-848
P2C
P3C
LPS

Example: Dendrogram

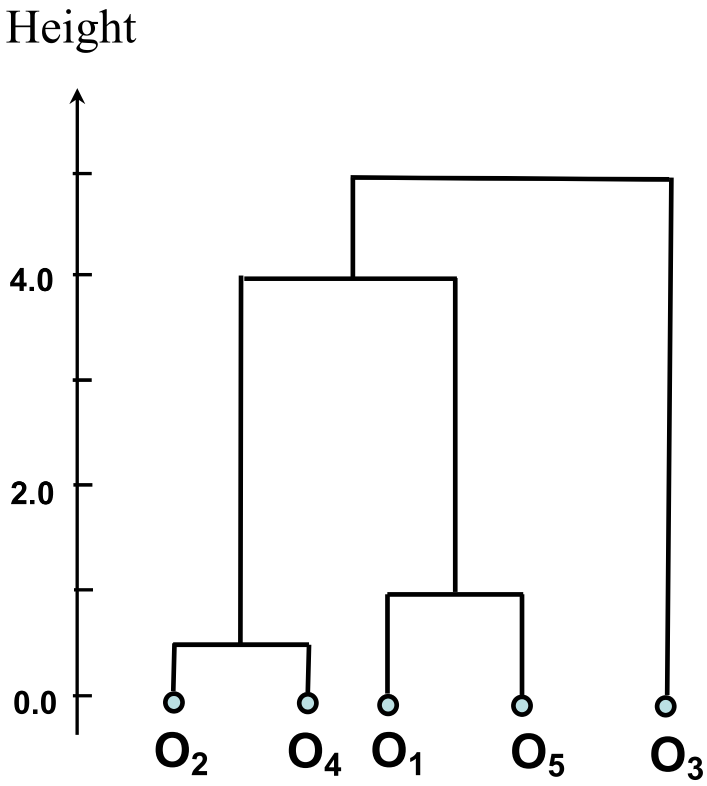
Based on the distance matrix, **D**, complete a clustering algorithm and corresponding dendrogram using single linkage clustering

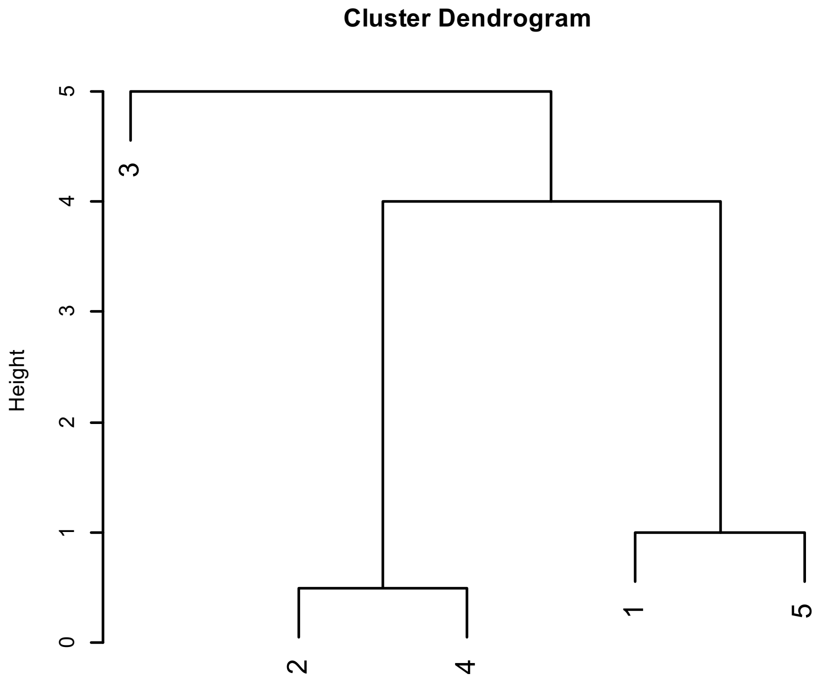
$$\mathbf{D} = \begin{bmatrix} 0 & 5 & 6 & 5.5 & 1 \\ 5 & 0 & 7 & 0.5 & 4 \\ 6 & 7 & 0 & 7.5 & 5 \\ 5.5 & 0.5 & 7.5 & 0 & 4.5 \\ 1 & 4 & 5 & 4.5 & 0 \end{bmatrix}$$

$$\begin{array}{c|cccc} & \mathbf{1} & \mathbf{\{2,4\}} & \mathbf{3} & \mathbf{5} \\ \hline \mathbf{1} & 0 & 5 & 6 & 1 \\ \mathbf{\{2,4\}} & 5 & 0 & 7 & 4 \\ \mathbf{3} & 6 & 7 & 0 & 5 \\ \mathbf{5} & 1 & 4 & 5 & 0 \end{array}$$

$$\begin{array}{c|ccc} & \mathbf{\{1,5\}} & \mathbf{\{2,4\}} & \mathbf{3} \\ \hline \mathbf{\{1,5\}} & 0 & 4 & 5 \\ \mathbf{\{2,4\}} & 4 & 0 & 7 \\ \mathbf{3} & 5 & 7 & 0 \end{array}$$

$$\begin{array}{c|cc} & \mathbf{\{1,2,4,5\}} & \mathbf{3} \\ \hline \mathbf{\{1,2,4,5\}} & 0 & 5 \\ \mathbf{3} & 5 & 0 \end{array}$$




Cluster Dendrogram

How to do this in R?

The basic R build has the `hclust` function which performs hierarchical clustering.

- The function output is an object of class **`hclust`** which describes the tree produced by the clustering process (and other information about how it was created).
- The resulting object can be plotted using the `plot` command to produce a dendrogram.
- The `hclust` object can also be used as input into the `cutree` function, which will divide the tree into clusters. The tree can be cut into either a requested number of clusters or at a requested dendrogram height.
- The syntax for `hclust` is

```
hclust(d, method = "complete", …)
```

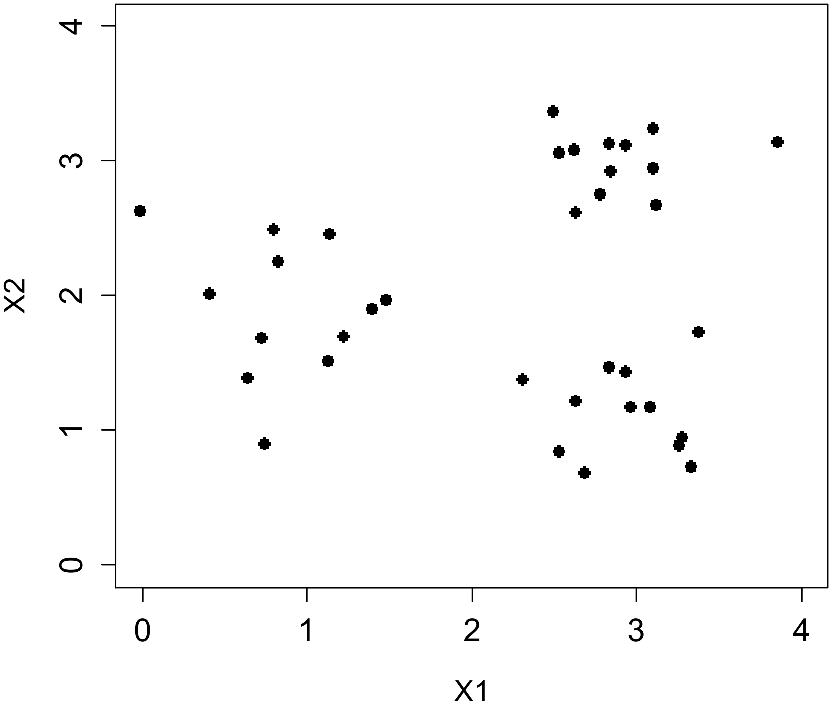
where `d` is a `dist` object and `method` is an agglomeration (linkage) method (the default is complete).

Cluster Example with in R

To demonstrate the use of the `hclust` function, we first examine the toy dataset `Cluster_Ex`, which has 36 units in it, each with two measurements: X1 and X2.

```
> head(Cluster_Ex)
              X1       X2
Unit 1 0.4102665 2.010689
Unit 2 3.0825080 1.175286
Unit 3 1.4751248 1.969292
Unit 4 2.6245358 3.080116
Unit 5 3.3782067 1.736558
Unit 6 2.4948872 3.361988
```

```
> plot(X2~X1,data=Cluster_Ex,xlim=c(0,4),
+ ylim=c(0,4),cex=1.3,cex.axis=1.3,pch=16)
```



A quick glance at the data set suggests that there could be 3 clusters in the dataset.

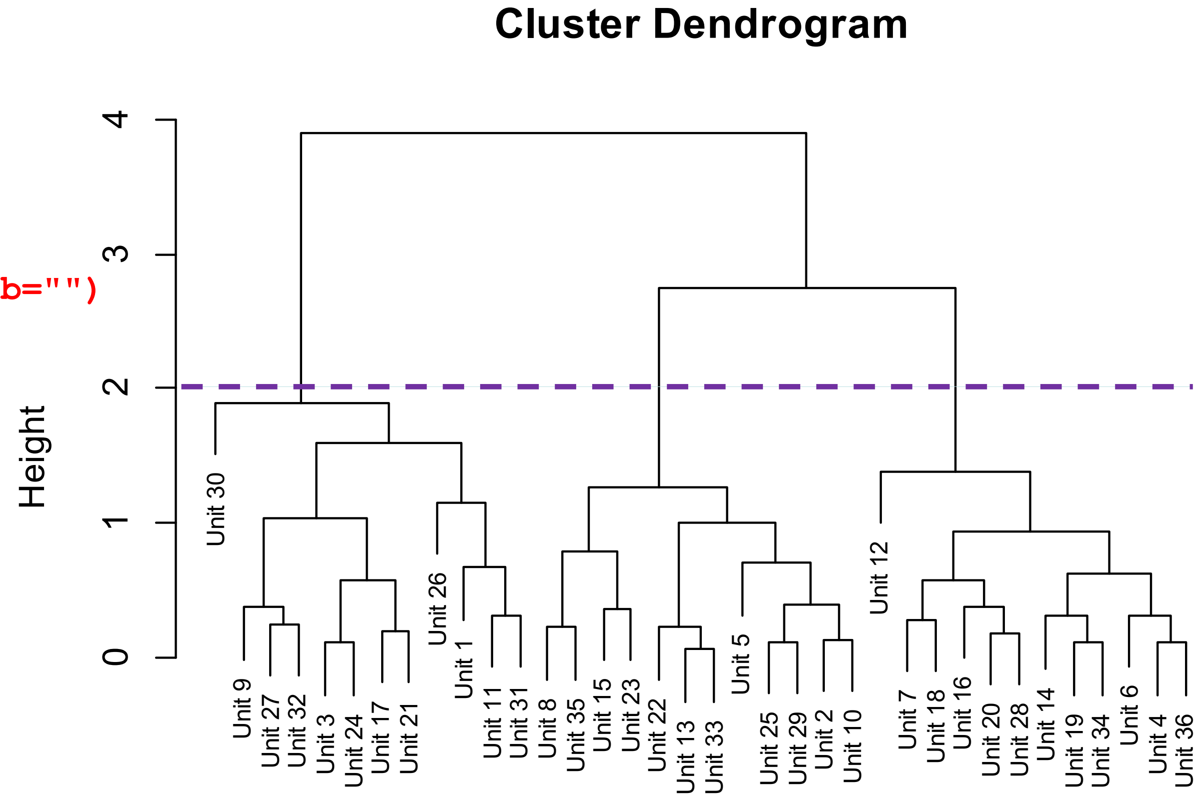
Cluster Example in R

```
> Cluster_Ex_HC<-hclust(dist(Cluster_Ex))
> Cluster_Ex_HC

Call:
hclust(d = dist(Cluster_Ex))

Cluster method   : complete
Distance         : euclidean
Number of objects: 36

> plot(Cluster_Ex_HC,cex=.75,xlab="")
```



We can see cutting the tree at a height of 2 would break the set into 3 clusters.

The labels on the leaves are the row names (or row numbers) from the data frame. Alternatively, we can provide a vector of labels when calling the `plot` function, i.e., `labels=names_vec`, or set `labels=F` to have no labels.

Cluster Example in R

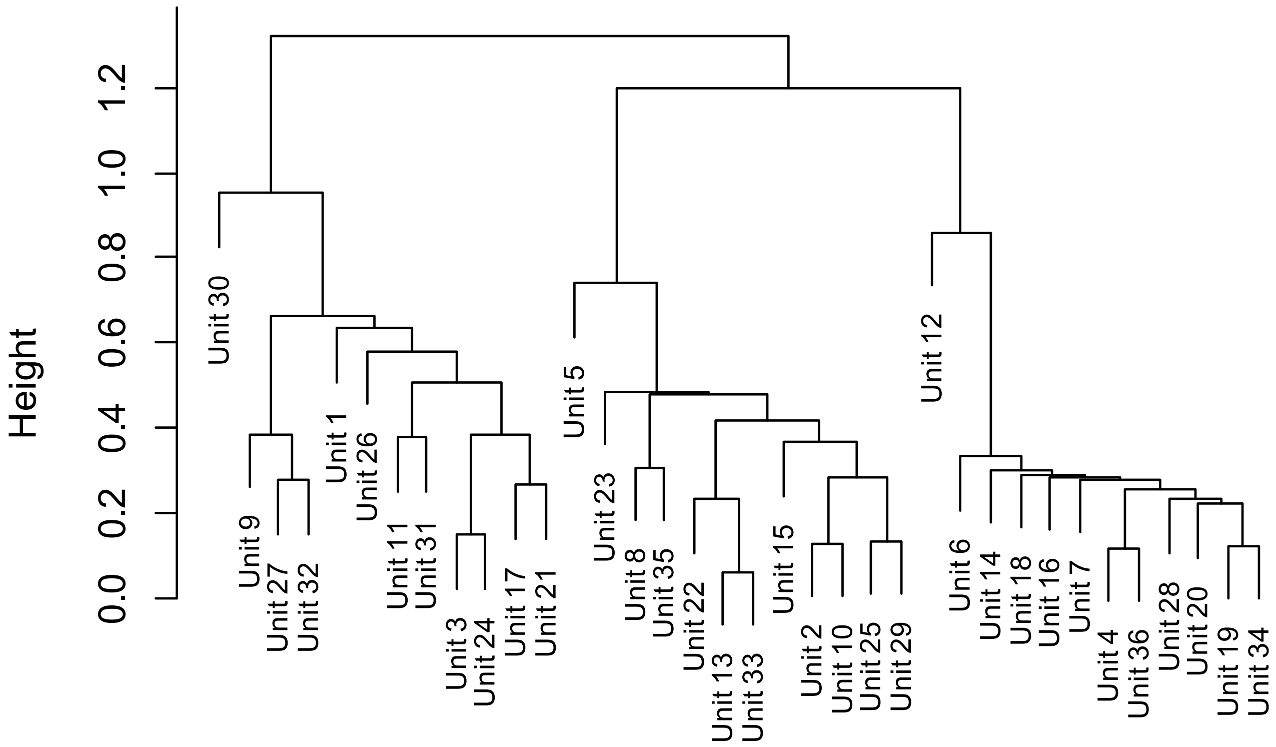
```
> Cluster_Ex_HC<-hclust(dist(Cluster_Ex,method="manhattan"),method="single")
> Cluster_Ex_HC

Call:
hclust(d = dist(Cluster_Ex, method = "manhattan"), method = "single")

Cluster method   : single
Distance         : manhattan
Number of objects: 36

> plot(Cluster_Ex_HC,cex=.75,xlab="")
```

Cluster Dendrogram



Cluster Example in R

An `hclust` object can be used as input into the `cutree` function, which will divide the tree into clusters. The tree can be cut into either a requested number of clusters or at a requested dendrogram height.

The syntax is cutree(tree, k = NULL, h = NULL) with arguments

tree: a tree as produced by hclust

k: an integer scalar or vector with the desired number of groups

h: numeric scalar or vector with heights where the tree should be cut.

`cutree` returns a vector with group memberships if `k` or `h` are scalar, otherwise a matrix with group memberships is returned where each column corresponds to the elements of `k` or `h`, respectively (which are also used as column names).

```
> Cluster_Ex_HC<-hclust(dist(Cluster_Ex))
> cutree(Cluster_Ex_HC,k=3)
 Unit 1  Unit 2  Unit 3  Unit 4  Unit 5  Unit 6  Unit 7  Unit 8  Unit 9 
      1       2       1       3       2       3       3       2       1 
Unit 10 Unit 11 Unit 12 Unit 13 Unit 14 Unit 15 Unit 16 Unit 17 Unit 18 
      2       1       3       2       3       2       3       1       3 
Unit 19 Unit 20 Unit 21 Unit 22 Unit 23 Unit 24 Unit 25 Unit 26 Unit 27 
      3       3       1       2       2       1       2       1       1 
Unit 28 Unit 29 Unit 30 Unit 31 Unit 32 Unit 33 Unit 34 Unit 35 Unit 36 
      3       2       1       1       1       2       3       2       3 
```

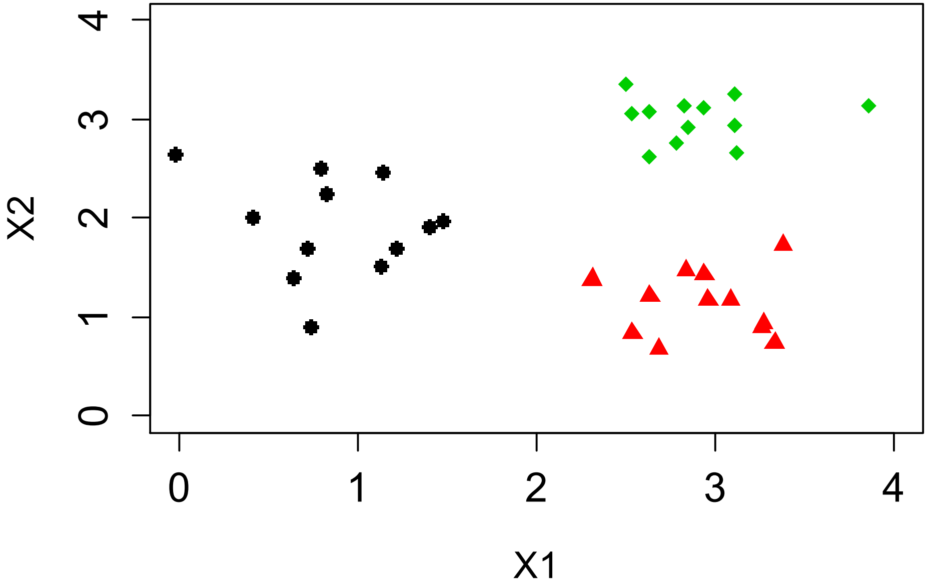
Cluster Example in R

```
> Cluster_Ex<-data.frame(Cluster_Ex,cutree(Cluster_Ex_HC,k=c(3,4,5)))
> names(Cluster_Ex)
[1] "X1" "X2" "X3" "X4" "X5"
> names(Cluster_Ex)<-c("X1","X2","k_3","k_4","k_5")
> head(Cluster_Ex)
              X1       X2 k_3 k_4 k_5
Unit 1 0.4102665 2.010689   1   1   1
Unit 2 3.0825080 1.175286   2   2   2
Unit 3 1.4751248 1.969292   1   1   3
Unit 4 2.6245358 3.080116   3   3   4
Unit 5 3.3782067 1.736558   2   2   2
Unit 6 2.4948872 3.361988   3   3   4
```

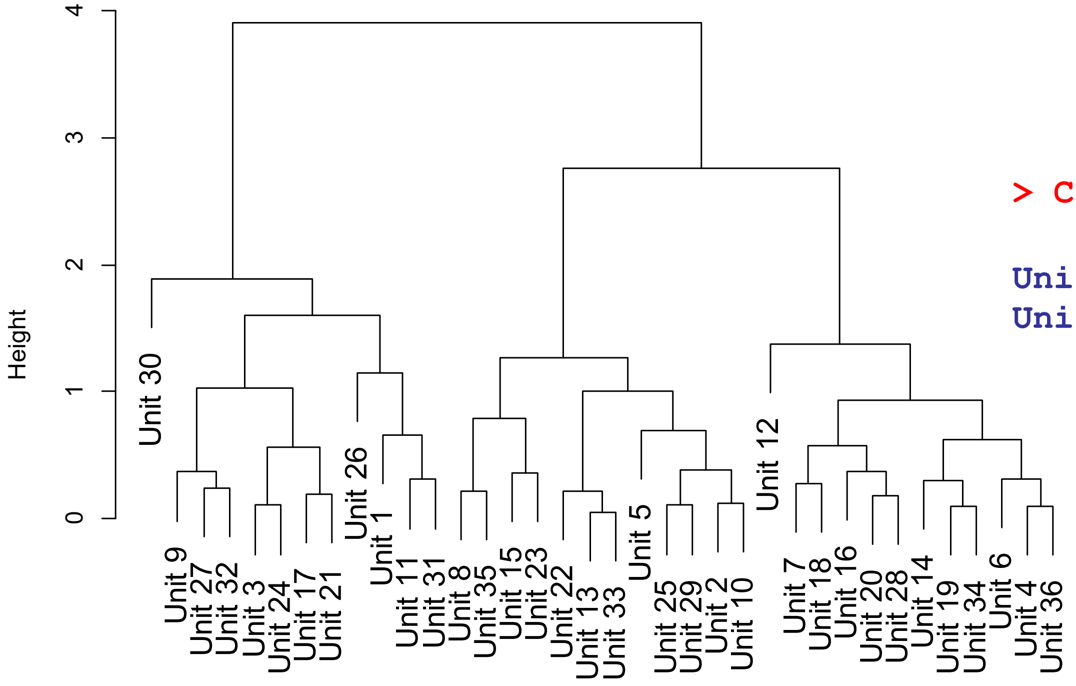
Entering the vector `k=c(3,4,5)` into the `cutree` function gives the cluster membership for the 3, 4, and 5 cluster solutions and we append these solutions onto the original data frame. We then rename these new columns with the number of clusters.

Cluster Example in R

```
> plot(X2~X1,data=Cluster_Ex,xlim=c(0,4),ylim=c(0,4),cex.axis=1.3,
+ cex.lab=1.2,cex=1.2,pch=15+k_3,col=k_3)
```



```
> plot(Cluster_Ex_HC,cex=1.25,xlab="")
```

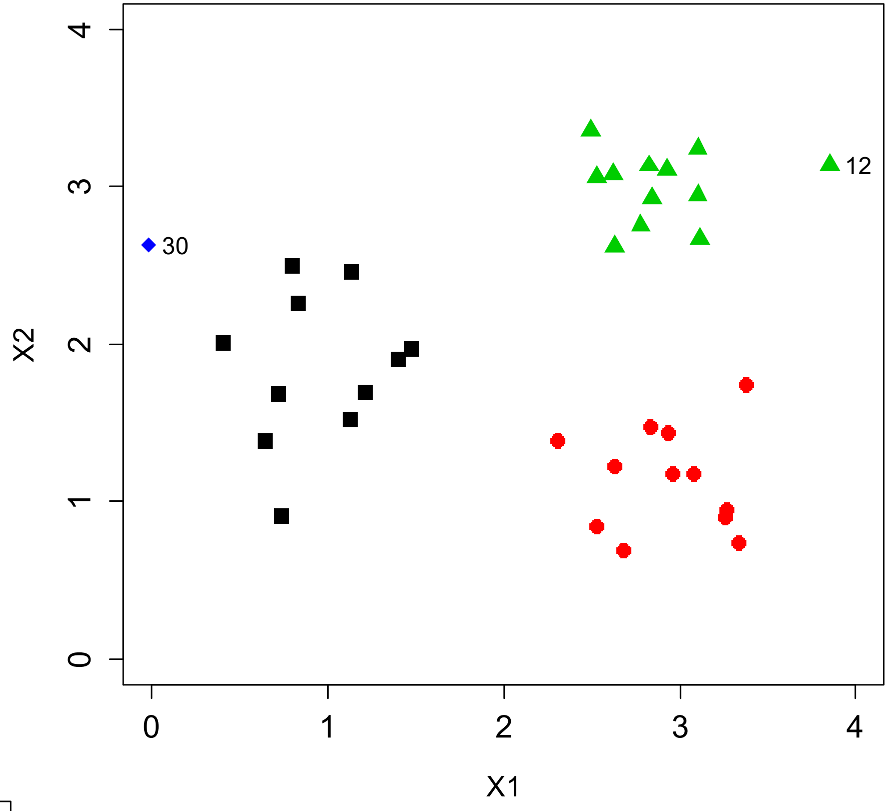


```
> Cluster_Ex[c(12,30),]
                 X1       X2 k_3 k_4 k_5
Unit 12  3.85290915 3.136639   3   3   4
Unit 30 -0.01895903 2.634556   1   4   5
```

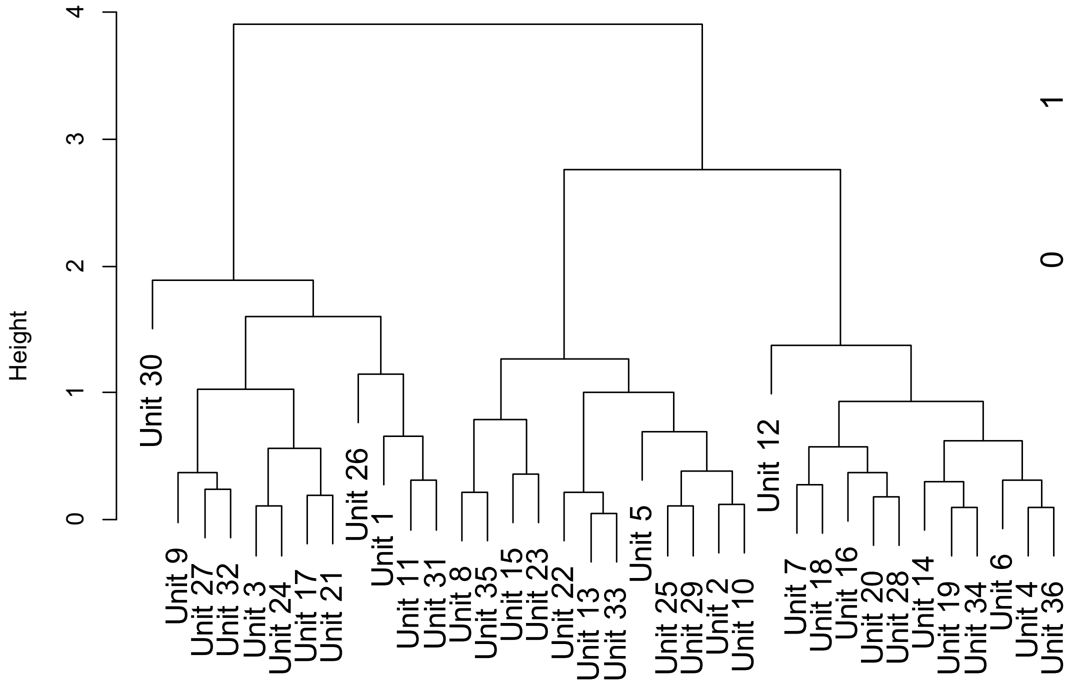
Cluster Example in R

```
> plot(X2~X1,data=Cluster_Ex,xlim=c(0,4),ylim=c(0,4),cex.axis=1.3,
+ cex=1.4,cex.lab=1.2,pch=14+k_4,col=k_4)
```

```
> identify(Cluster_Ex)
```



```
> plot(Cluster_Ex_HC,cex=1.25,xlab="")
```



Cluster Example in R

Plotting options for an `hclust` object.

1. You can change the width or color of the dendrogram lines by submitting the `col` or `lwd` parameters, respectively, with the plot function.

2. You can add a title (or subtitle) using the `main` or `sub` arguments as with other plots.

3. You can adjust the length of the terminal edges using the `hang` argument. This is the fraction of the plot height by which labels should hang below the rest of the plot. A negative value will cause the labels to hang down from 0.

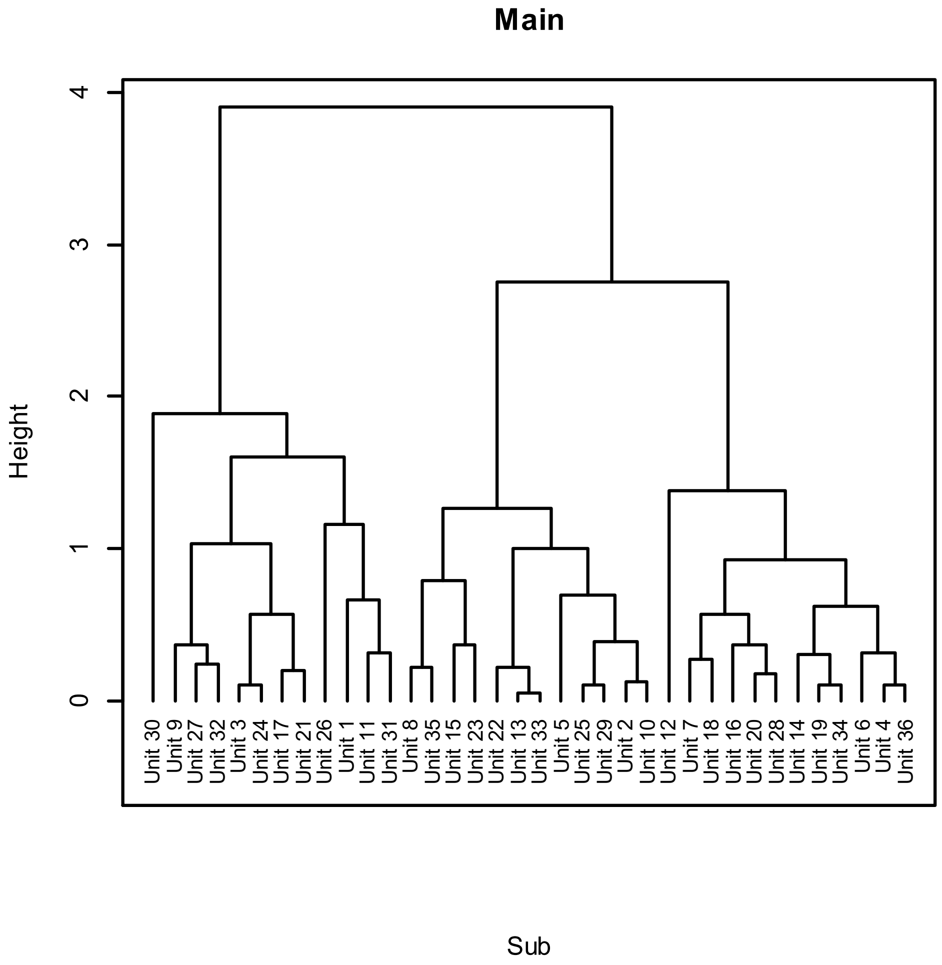
4. You can draw a box around the plot setting `frame.plot=T` when calling the plot function or by using the `box()` function after generating the plot.

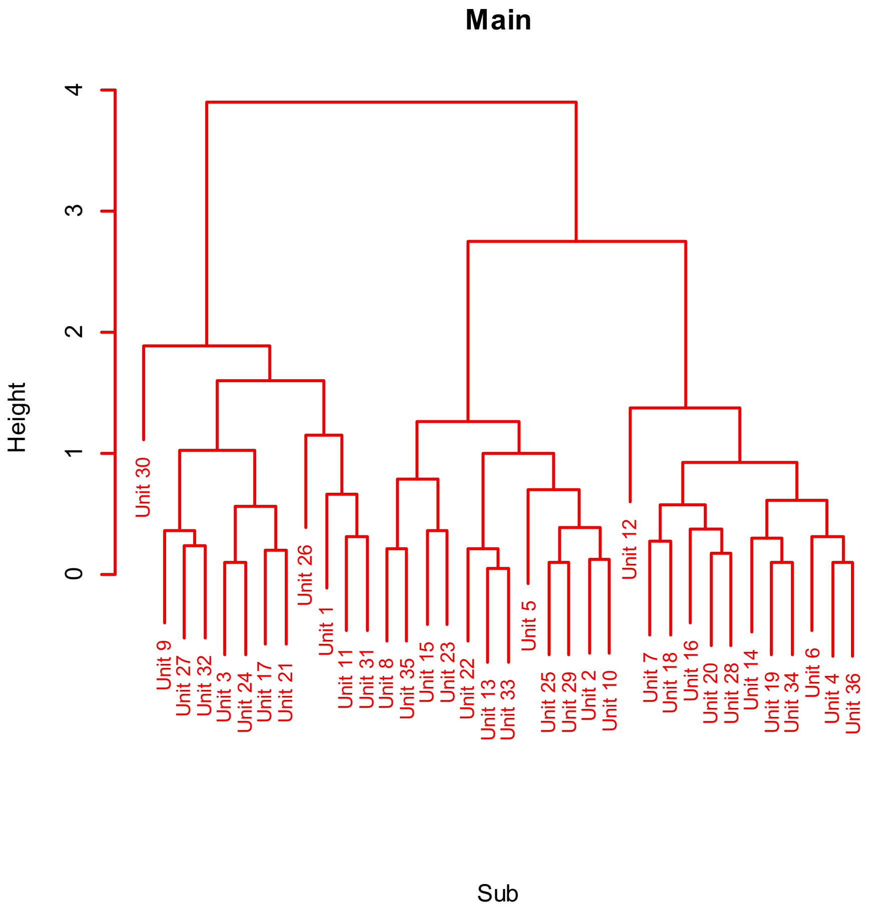
5. The `cex` argument controls the label size.

Type `?hclust` for more information on the `hclust` function in R.

Examples Plotting an `hclust` Object

```
> plot(Cluster_Ex_HC,hang=-1,main="Main",sub="Sub",lwd=2,cex=.8,xlab="",frame.plot=T)
> plot(Cluster_Ex_HC,hang=.2,main="Main",sub="Sub",lwd=2,cex=.8,col="red2",xlab="")
```

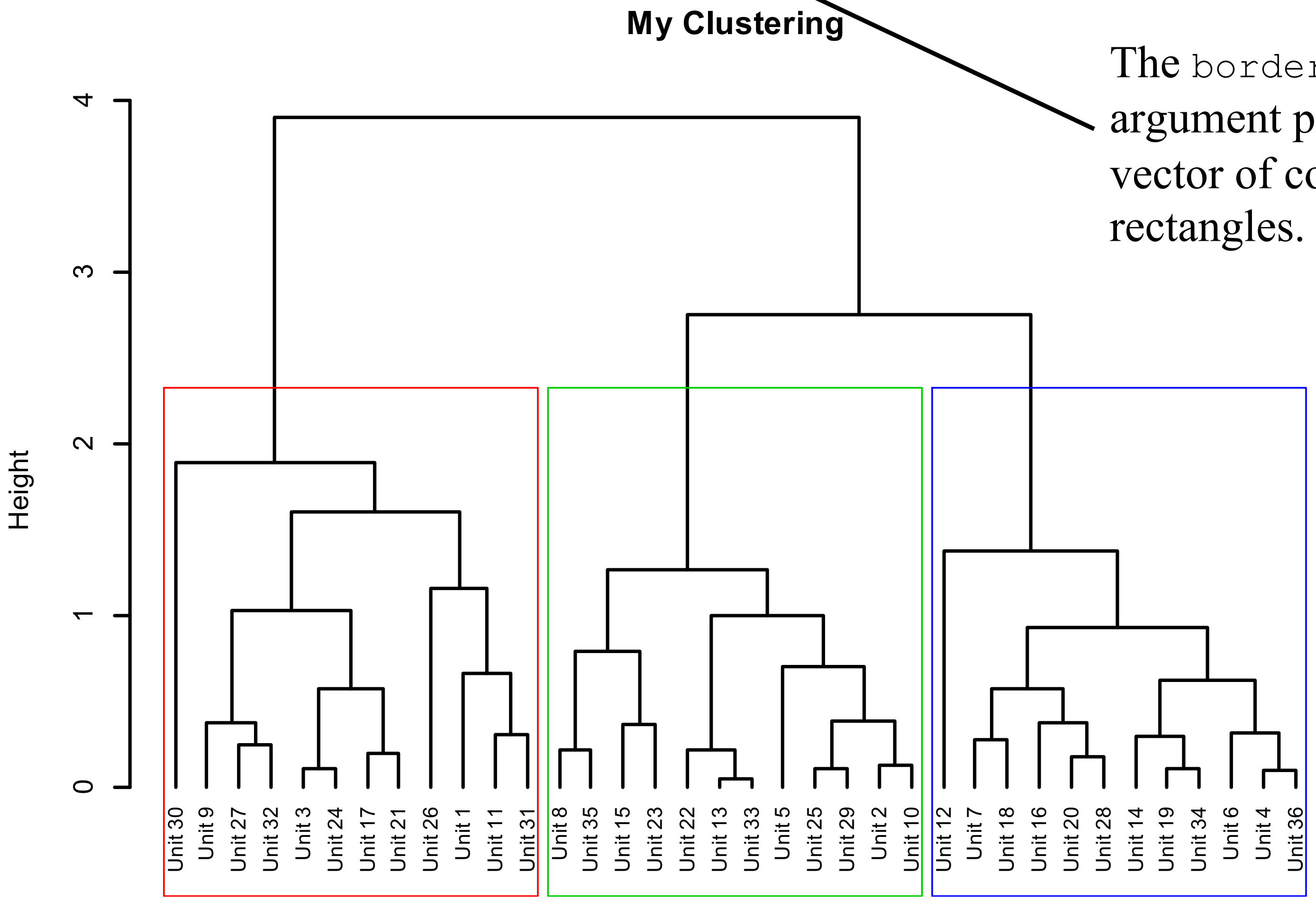




Adding Rectangles Around Clusters

You can add rectangles around a set of clusters using the `rect.hclust` function after plotting the dendrogram. The syntax is similar to the `cutree` function.

```
> plot(Cluster_Ex_HC,hang=-1,lwd=2,cex=.75,xlab="",main="My Clustering",sub="")
> rect.hclust(Cluster_Ex_HC,k=3,border=2:4)
```

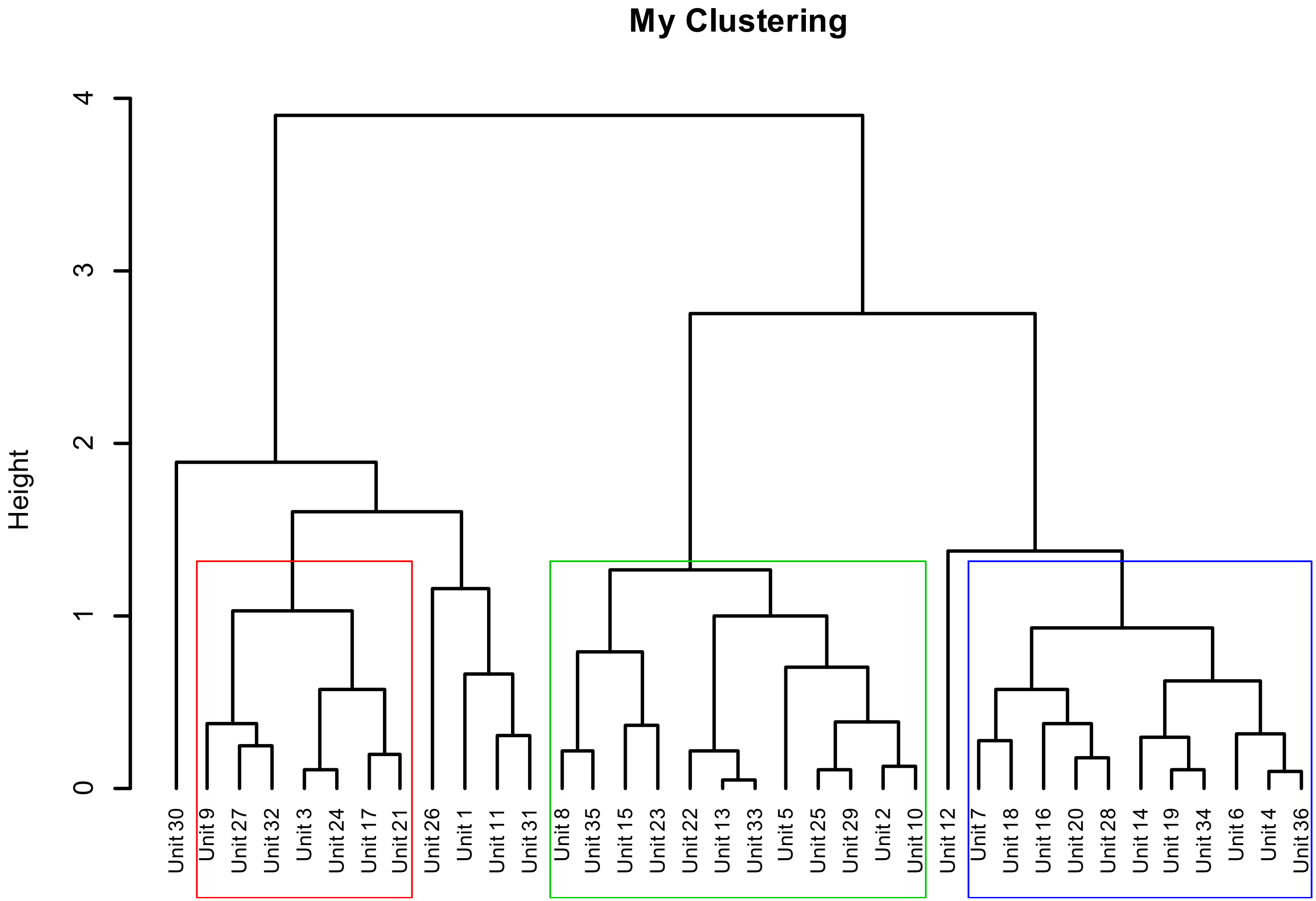


The `border` (optional) argument provides a vector of colors for the rectangles.

Adding Rectangles Around Clusters

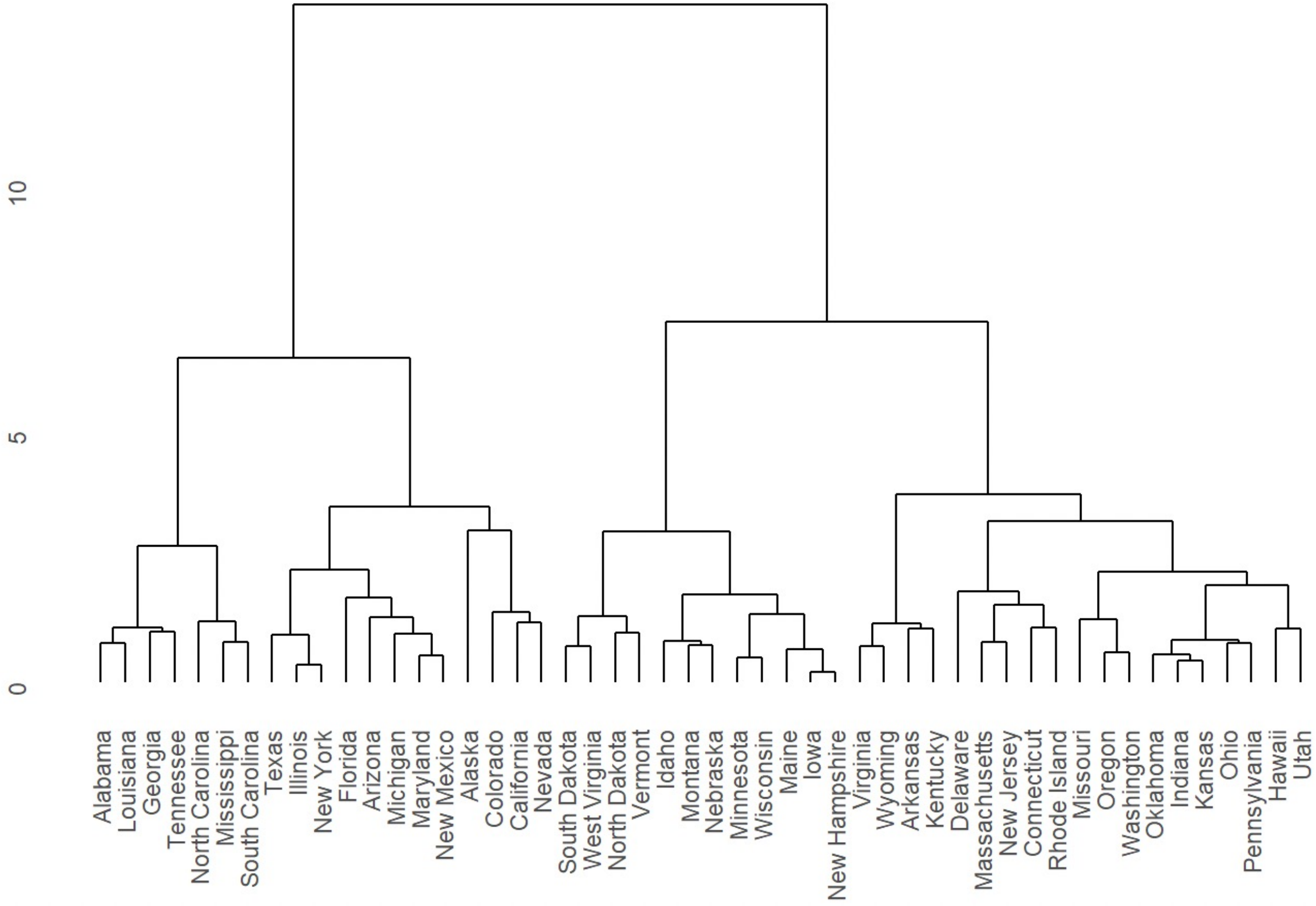
The `which` optional argument in the `rect.hclust` function can be used to select which clusters to outline (from left to right).

```
> plot(Cluster_Ex_HC,hang=-1,lwd=2,cex=.75,xlab="",main="My Clustering",sub="")
> rect.hclust(Cluster_Ex_HC,k=6,which=c(2,4,6),border=2:4)
```



ggplot2 implementation: ggdendro package

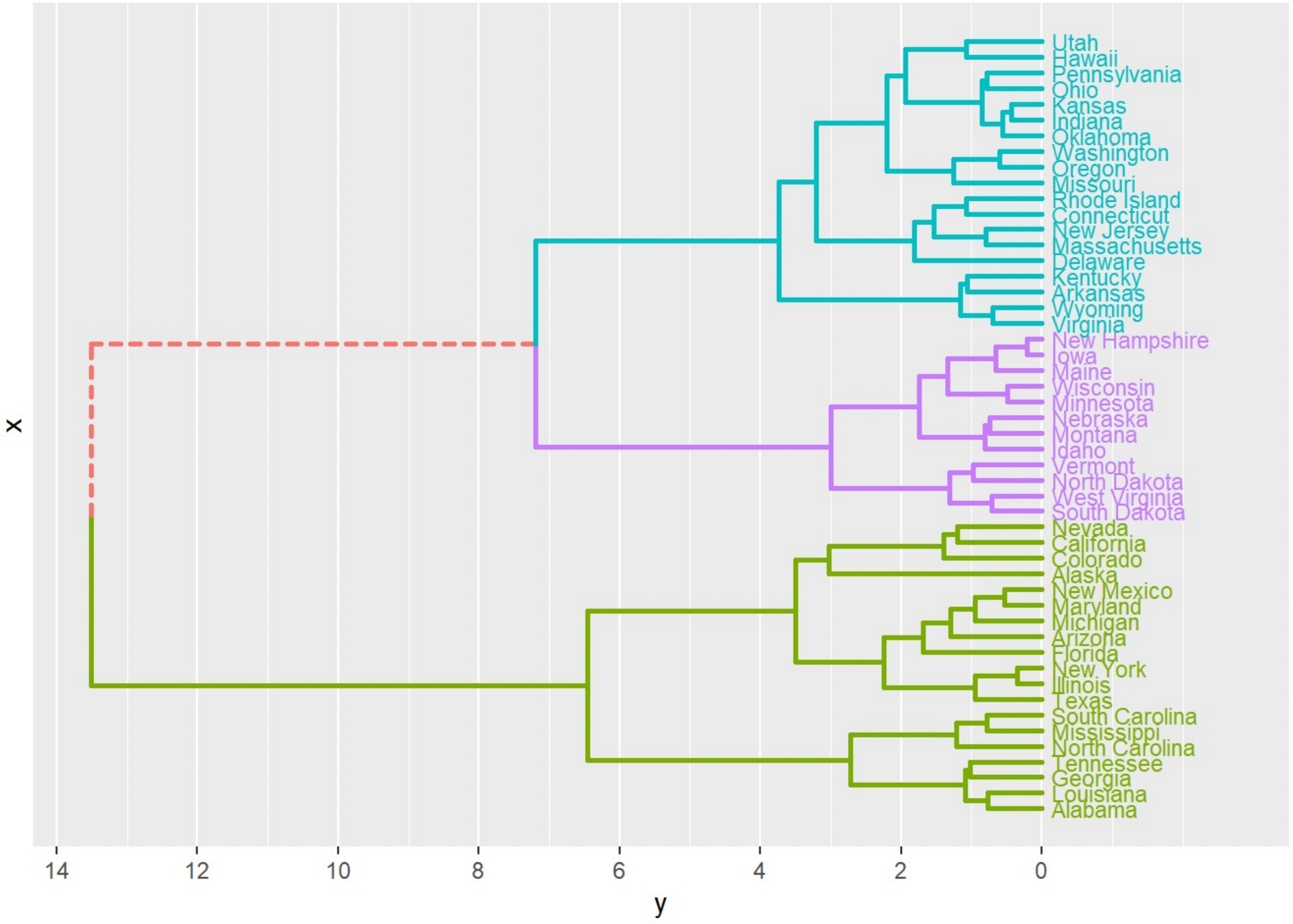
```
➢ data(USArrests) # pre-installed data
➢ dd <- dist(scale(USArrests), method = "euclidean")
➢ hc <- hclust(dd)
➢ ggdendrogram(hc)
```



ggplot2 implementation: ggdendro package

- `hcdata <- dendro_data_k(hc, 3)`
- `plot_ggdendro(hcdata, direction = "lr", expand.y = 0.2) # try fan = TRUE`

Check github notes for the functions used on this plot!


Utah
Hawaii
Pennsylvania
Ohio
Kansas
Indiana
Oklahoma
Washington
Oregon
Missouri
Rhode Island
Connecticut
New Jersey
Massachusetts
Delaware
Kentucky
Arkansas
Wyoming
Virginia
New Hampshire
Iowa
Maine
Wisconsin
Minnesota
Nebraska
Montana
Idaho
Vermont
North Dakota
West Virginia
South Dakota
Nevada
California
Colorado
Alaska
New Mexico
Maryland
Michigan
Arizona
Florida
New York
Illinois
Texas
South Carolina
Mississippi
North Carolina
Tennessee
Georgia
Louisiana
Alabama
x
y