Week 9 Clustering

Unit Co-ordinator: Dr. Liwan Liyanage

School of Computing, Engineering and Mathematics

WESTERN SYDNEY
UNIVERSITY

Cluster Analysis

- Cluster Analysis or Clustering is a form of ***unsupervised learning***. There is no specific response variable.
- The aim is to find groups in data; i.e. group observations in a data set into clusters with similar values of their variables.

We will look at

- k-means clustering
- hierarchical clustering
- distance metrics
- practical issues

WESTERN SYDNEY UNIVERSITY

Cluster Analysis

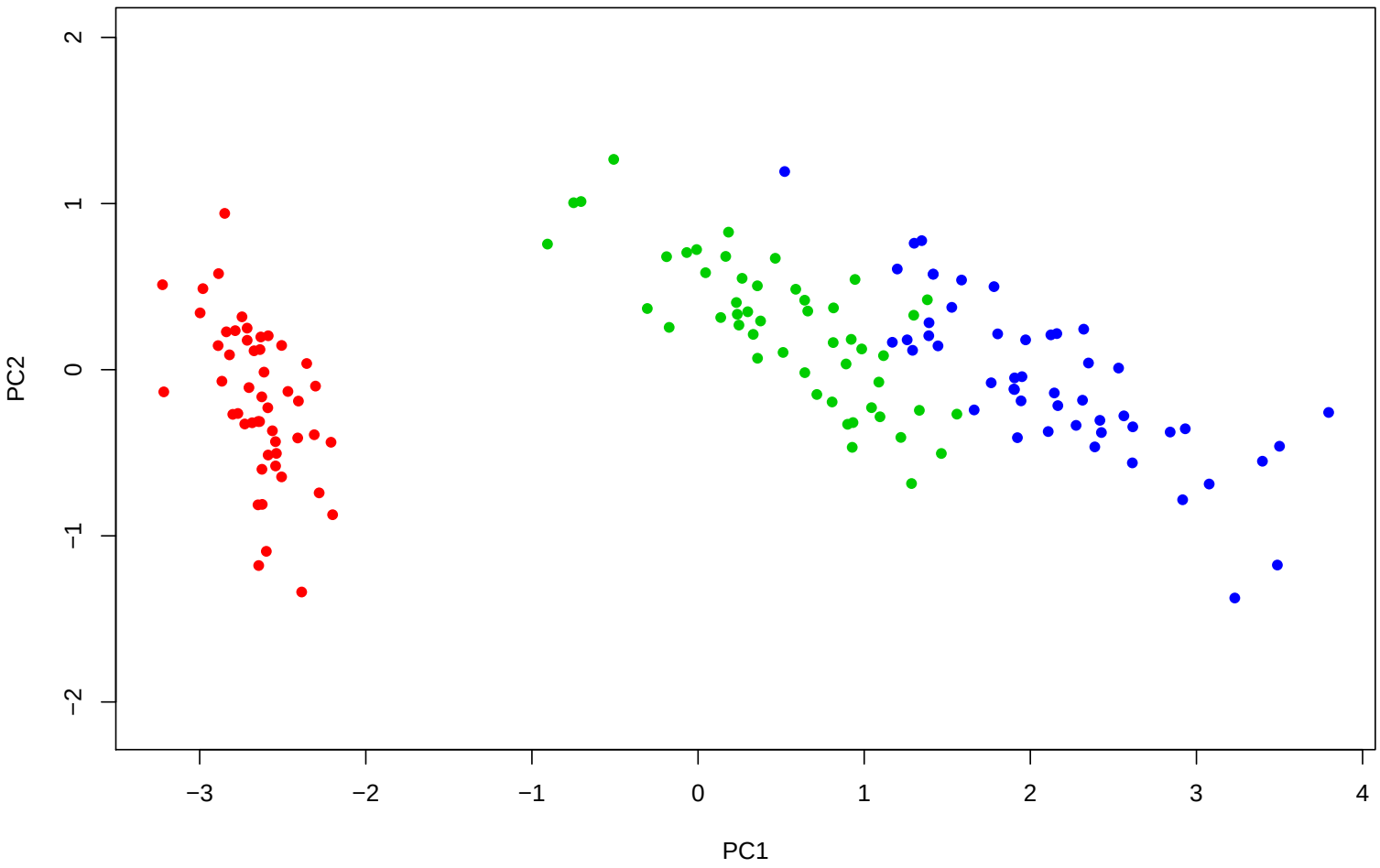
Like Principle Components Analysis, Clustering is about simplifying a data set. PCA aims to reduce the dimensions of data set so that we can better see patterns without losing too much information.

Cluster Analysis seeks to find groups in a data set directly, where members of groups are more similar to members of the same group than other groups.

We need to define what it means to be a group and what it means to be similar.

WESTERN SYDNEY UNIVERSITY

Clusters



WESTERN SYDNEY UNIVERSITY

Similarity

In fact, we work with dissimilarity or distance. The simplest and most commonly used dissimilarity is ***Euclidean distance***. Euclidean distance is what we mean by distance when we measure it with a ruler or tape measure.

Euclidean distance between points **A** and **B** is:

- always zero or greater
- zero if and only if two points are the same (in the same place)
- less than the sum of the distances from **A** to **C** and **C** to **B** (*Triangle Inequality*)

WESTERN SYDNEY UNIVERSITY

Euclidean Distance

If X is a data matrix, i.e. X_{ij} is the value of the j^{th} variable measured on the i^{th} individual, then Euclidean distance between the i^{th} and i'^{th} is;

$$d(X_i - X_{i'}) = \sqrt{\sum_{j=1}^{p}(X_{ij} - X_{i'j})^2}$$

with ***only 2 measurements*** (dimensions).

cf. Pythagoras Theorem

WESTERN SYDNEY UNIVERSITY

k-means

The idea of k-means clustering is *to seek groupings or clusters so that the distance between points within clusters is as small as possible.* Although, in fact, the within cluster sum of squares is ***minimized***.

$$\sum_{(clusters)} \sum_{(i,i' in clusters)} d(X_i - X_{i'})^2$$

The set of clusters is chosen to minimize this sum.

Cluster Centroid

A cluster centroid is the centre of a cluster. It is found by averaging each variable for all the observations in the cluster.

WESTERN SYDNEY UNIVERSITY

k-means

Start with a desired number of clusters k

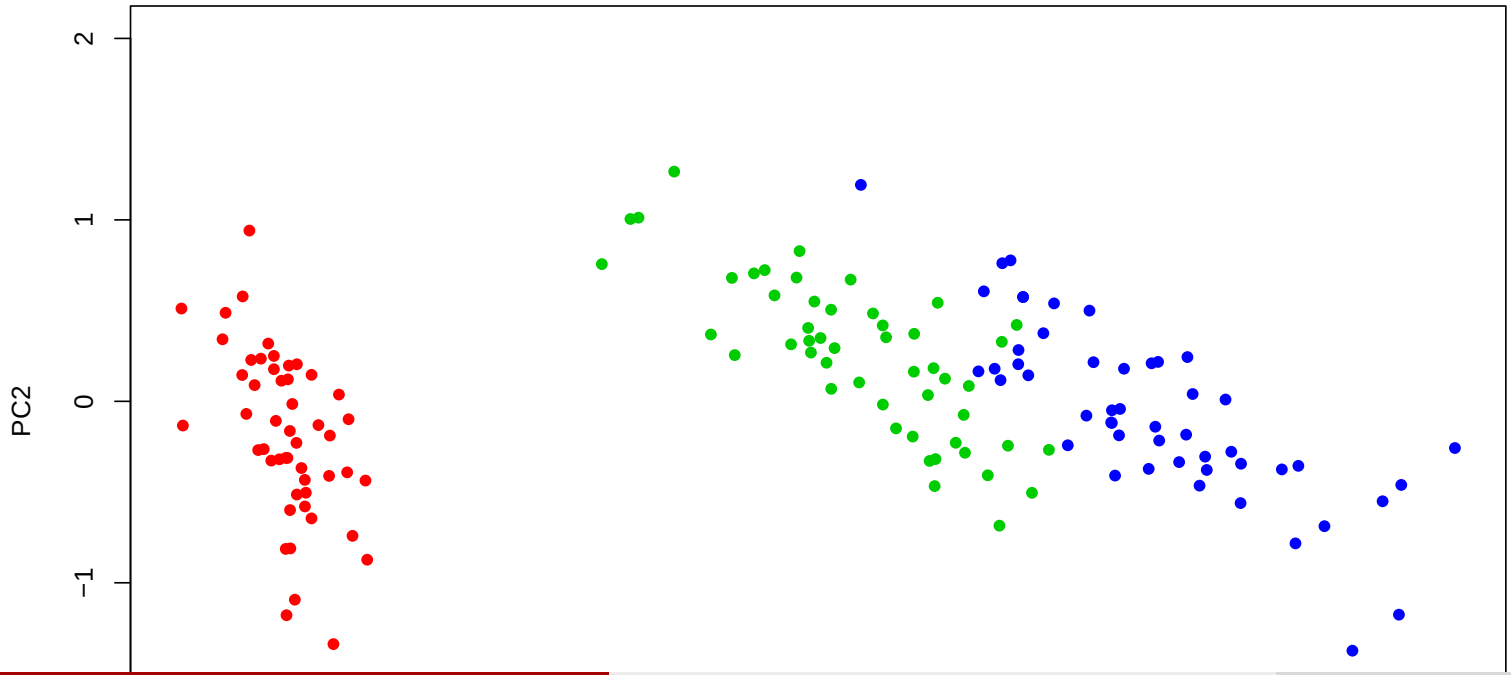
1. Randomly assign a number between 1 and k to each point. These are initial cluster labels.
2. For each cluster, find the centroid.
3. Assign each observation a new cluster based on the centroid it is closest to.
4. Repeat 2 and 3 above until the centroids (or clusters) don't change.

WESTERN SYDNEY UNIVERSITY

Iris Data - Plot PC1 against PC2

Colours represented by Species

```
 obj = prcomp(iris[,1:4])
plot(obj$x[,1:2], col=unclass(iris$Species)+1,
     pch=16, asp=1)
```



WESTERN SYDNEY UNIVERSITY

k-means Clustering

k-means needs us to specify the number of clusters to seek. Example k=3

```
X = iris[,1:4]
km = kmeans(X, centers = 3)
```

WESTERN SYDNEY UNIVERSITY

Random Start Function (*nstart*)

There are additional arguements for maximum number of iterations (*iter.max*) and the random start number (*nstart*).

```
km2 = kmeans(iris[,1:4], 3, nstart = 20, iter.max=100)
km2$cluster
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3 3 3 3 3 3 3
##  [71] 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [106] 2 3 2 2 2 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2
## [141] 2 2 3 2 2 2 3 2 2 3
```

The Cluster Allocations

The cluster allocations can be obtained using the function (*fitted*).

```
fitted(km, "classes")
```

```
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [106] 3 1 3 3 3 3 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3
## [141] 3 3 1 3 3 3 1 3 3 1
```

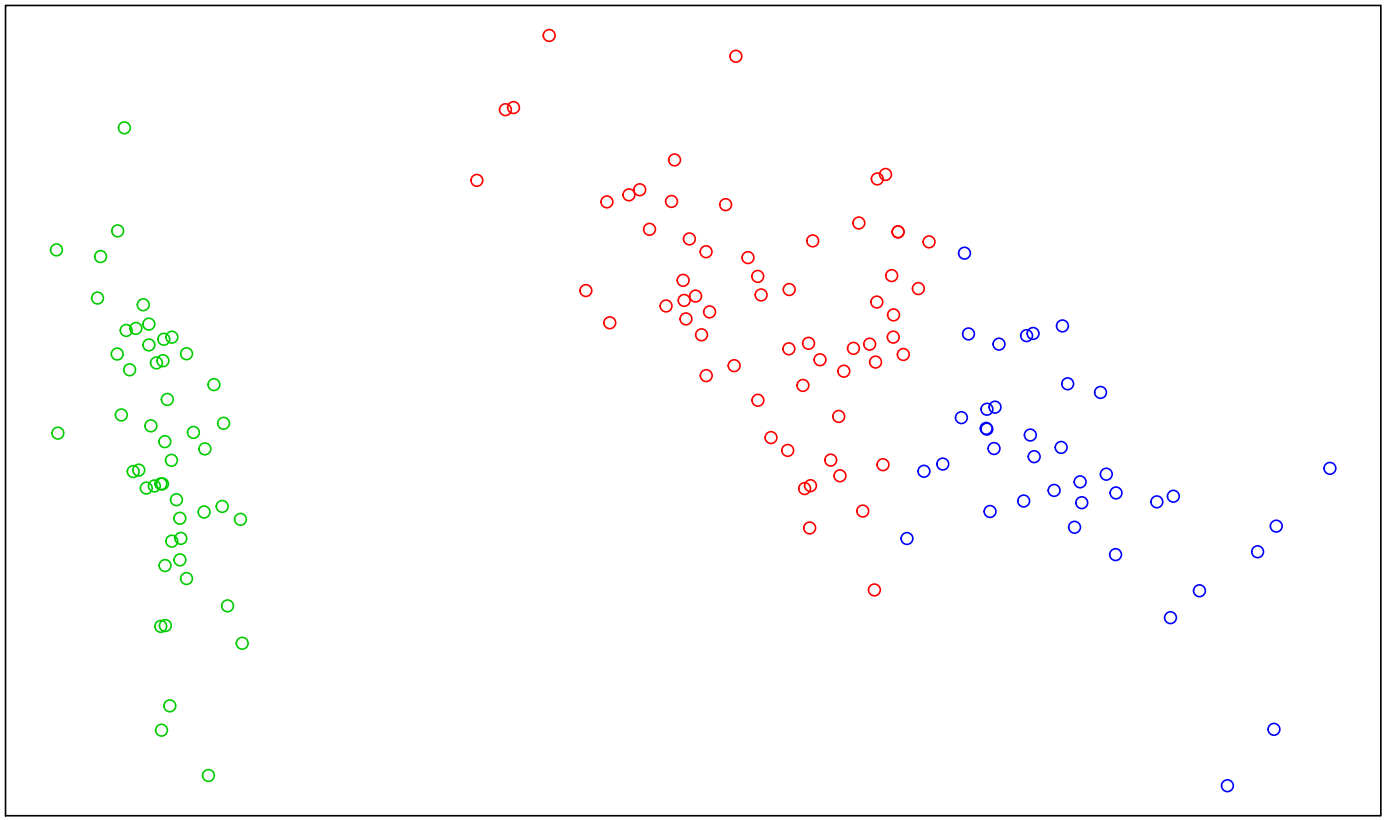
Plotting k-means Clusters

The colours indicate the observations according to the k-means cluster label depicting k-means clusters. Not according to the original variable Species

```
pp = prcomp(X)
plot(pp$x[,1:2], col=fitted(km, "classes")+1,
     xaxt="n", yaxt="n")
```

WESTERN SYDNEY UNIVERSITY

Plotting k-means (Continued...)



WESTERN SYDNEY UNIVERSITY

Comparing Cluster Lables with Species

We can now compare the k-means cluster with the true species.

```
table(species=iris$Species, cluster=fitted(km, "classes"))
```

```
##                cluster
## species          1  2  3
##   setosa         0 50  0
##   versicolor    48  0  2
##   virginica     14  0 36
```

WESTERN SYDNEY UNIVERSITY

km Output

```
km
```

```
## K-means clustering with 3 clusters of sizes 62, 50, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.901613    2.748387     4.393548    1.433871
## 2     5.006000    3.428000     1.462000    0.246000
## 3     6.850000    3.073684     5.742105    2.071053
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [106] 3 1 3 3 3 3 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3
## [141] 3 3 1 3 3 3 1 3 3 1
```

km Output Continued

```
km$iter
```

```
## [1] 2
```

```
km$ifault
```

```
## [1] 0
```

```
km$size
```

```
## [1] 62 50 38
```

```
km$betweenss
```

```
## [1] 602.5192
```

```
km$tot.withinss
```

Hierarchical Clustering

- k-means implicitly uses Euclidean distance
- The data must be all *quantitative measurements*
- If some data is *qualitative*, more advanced algorithms exist
- hierarchical clustering uses an arbitrary distance

There are two types of hierarchical clustering

- ***agglomerative***
- ***divisive***

We discuss the first method here.

WESTERN SYDNEY UNIVERSITY

Agglomerative Hierarchical Clustering

Suppose we have a possibly high dimensional data set X, and a distance defined between observations; e.g. Euclidean distance. The idea of agglomerative hierarchical clustering, is *to gradually merge clusters together to get a hierarchy of cluster solutions.*

1. Start with all observations in their own clusters - therefore, n clusters.
2. Merge the closest 2 clusters, to produce 1 fewer cluster.
3. Repeat 2, until all observations are in 1 cluster.

WESTERN SYDNEY UNIVERSITY

Agglomerative Hierarchical Clustering (Continued...)

When the clusters contain more than one point, we have to consider what closest means.

How to measure the distance between two clusters **A** and **B**;

- The ***minimum*** of distances between points in **A** and points in **B**.
- The ***average*** of distances between points in **A** and points in **B**.
- The ***maximum*** of distances between points in **A** and points in **B**.
- Anything else.

Each of these definitions produces a different clustering methods.

WESTERN SYDNEY UNIVERSITY

Distance Metrics

The most commonly used distance is ***Euclidean distance***. The distance between x and y is then;

$$\sqrt{\sum_j (y_j - x_j)^2}$$

Alternatives are,

- the ***City Block*** or ***Manhattan distance*** (The sum of the distances on each variable):

$$\sum_j (y_j - x_j)$$

- the ***Maximum distance***:

$$max_j (y_j - x_j)$$

WESTERN SYDNEY UNIVERSITY

Distance Metrics (Continued...)

For binary data (e.g. presence/ absence), the binary metrics is often used. Suppose you have data where x and y are sets of indicators of the presence of somethings. E.g.

- Shopping baskets : each variable is "1" if an item is being bought
- Archaelogical sites : "0/1" indicates artefact found or not.
- Documents : "0/1" indicates word present or not.

WESTERN SYDNEY UNIVERSITY

Distance Metrics (Continued...)

The binary metric compares two observations by looking at the number of things in common versus those not in common

$$d(x, y) = A/(A + B + C)$$

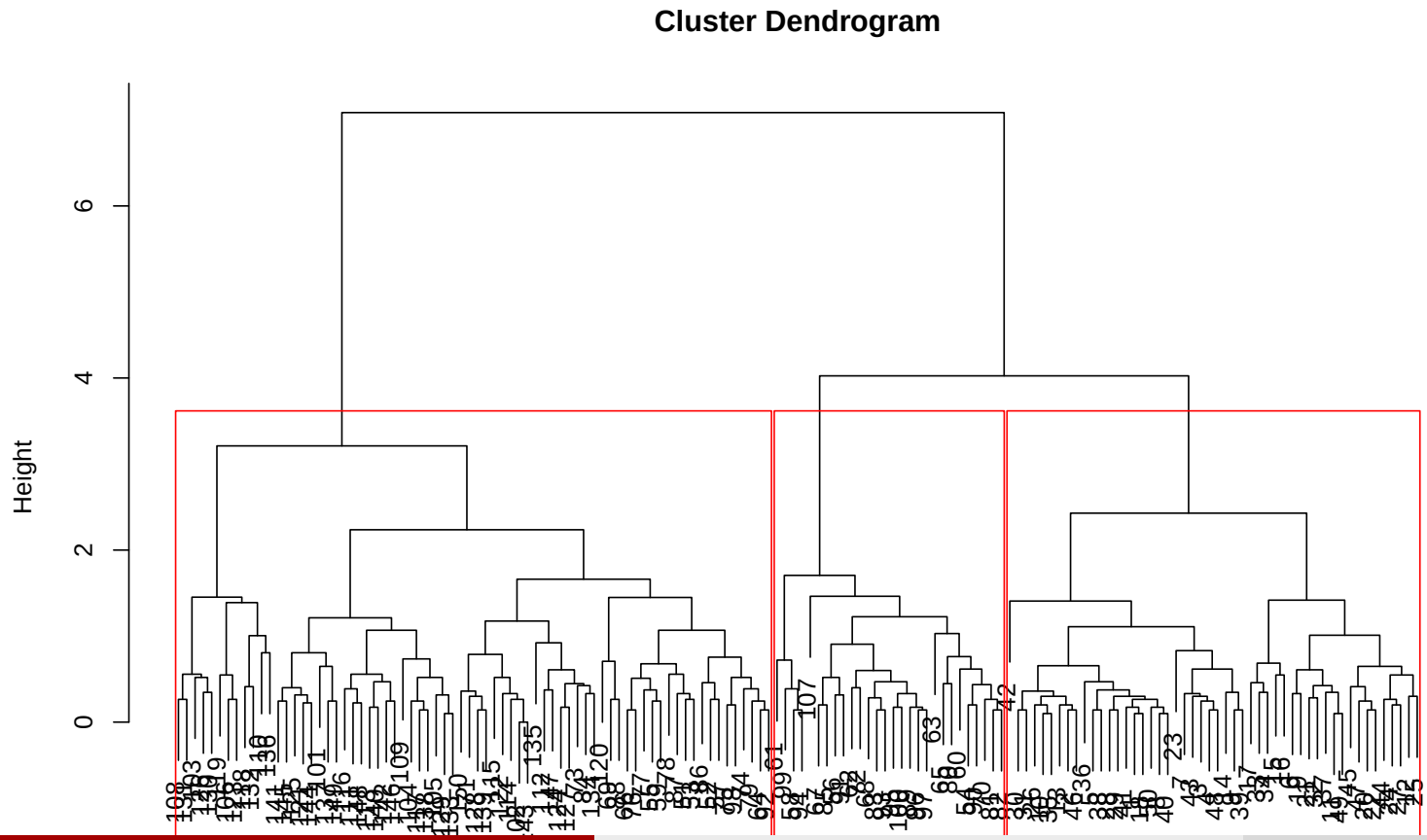
where

- A : items in both x and y
- B : items in x only
- C : items in y only
- D : items not in either x or y (not used)

WESTERN SYDNEY UNIVERSITY

Examples Hierarchical Clustering

```
hh = hclust(dist(X))
plot(hh, main = "Cluster Dendrogram")
rect.hclust(hh, k=3)
```



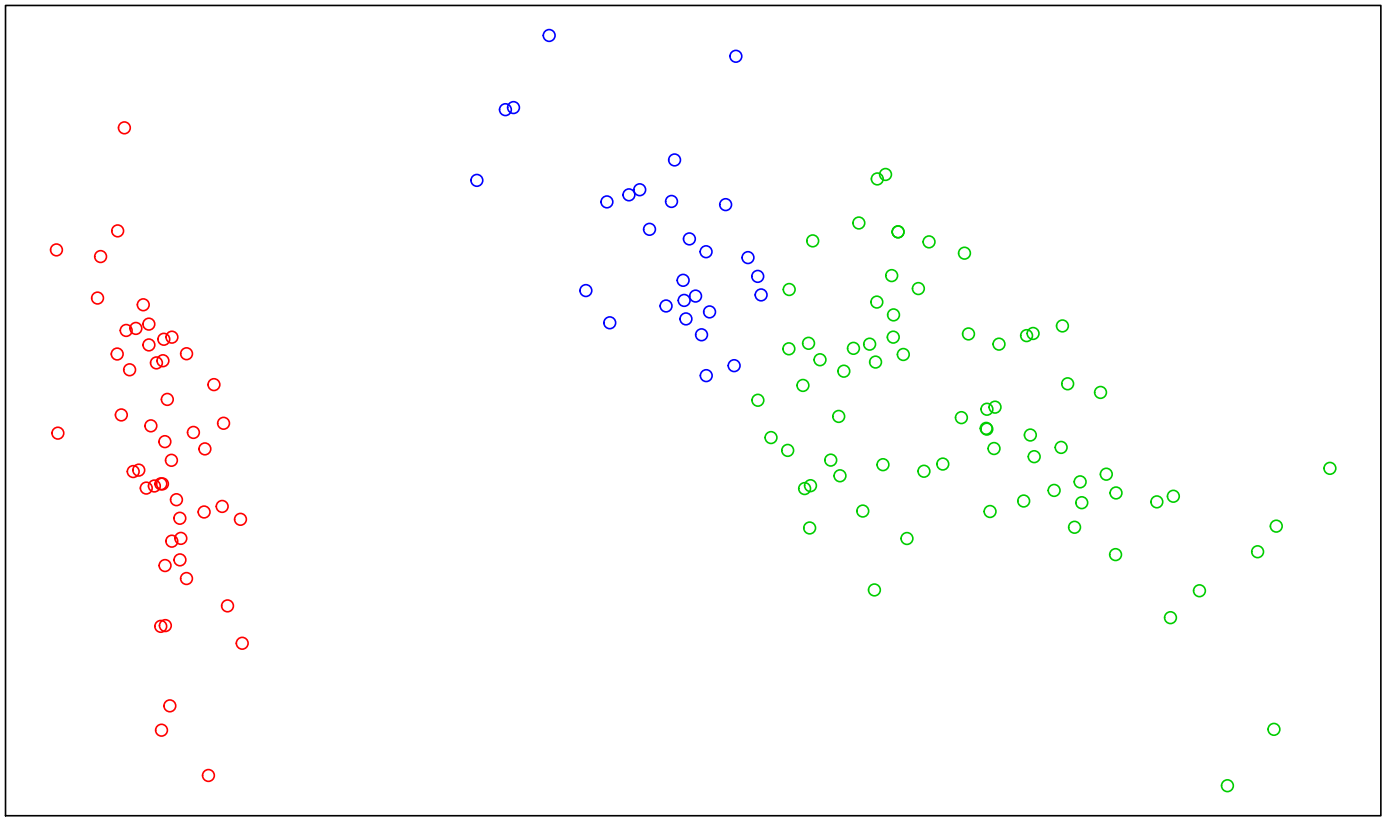
WESTERN SYDNEY UNIVERSITY

Examples Hierarchical Clustering (Continued...)

is a ***cluster dendrogram***; the height represents the ditance at which clusters were merged.

WESTERN SYDNEY UNIVERSITY

Examples Hierarchical Clustering - Iris Data



WESTERN SYDNEY UNIVERSITY

Examples Hierarchical Clustering - Iris Data (Continued...)

We apply *dist* first to get distances, then *hclust*. The result has the solution for all cluster numbers.

```
hh = hclust(dist(X), method="complete")
```

method can be "single", "average" and "complete" (plus others) the distance between two clusters **A** and **B** is

- ***single*** - The minimum of distances between points in **A** and points in **B**.
- ***average*** - The average of distances between points in **A** and points in **B**.
- ***complete*** - The maximum of distances between points in **A** and points in **B**.

Cluster Membership

To extract cluster membership, we have to decide how many clusters.

```
cutree(hh, k=3)
```

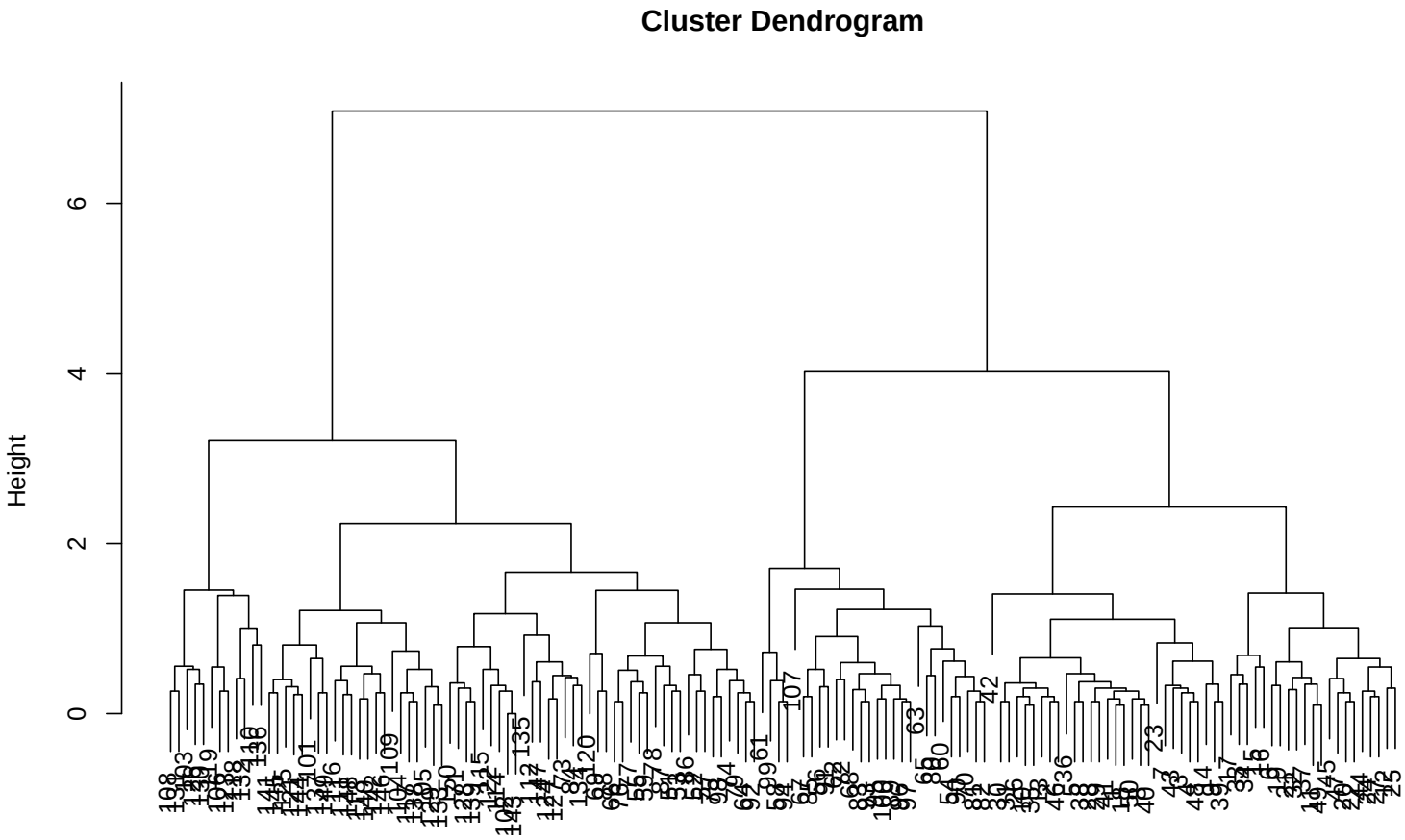
```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 3 2 3 2 3 2 3 3 3
##  [71] 2 3 2 2 2 2 2 2 2 3 3 3 3 2 3 2 2 2 3 3 3 2 3 3 3 3 3
## [106] 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [141] 2 2 2 2 2 2 2 2 2 2
```

WESTERN SYDNEY UNIVERSITY

Plotting

We use plot to get the dendrogram.

```
plot(hh, xlab=" ", sub="Complete link cluster analysis")
```

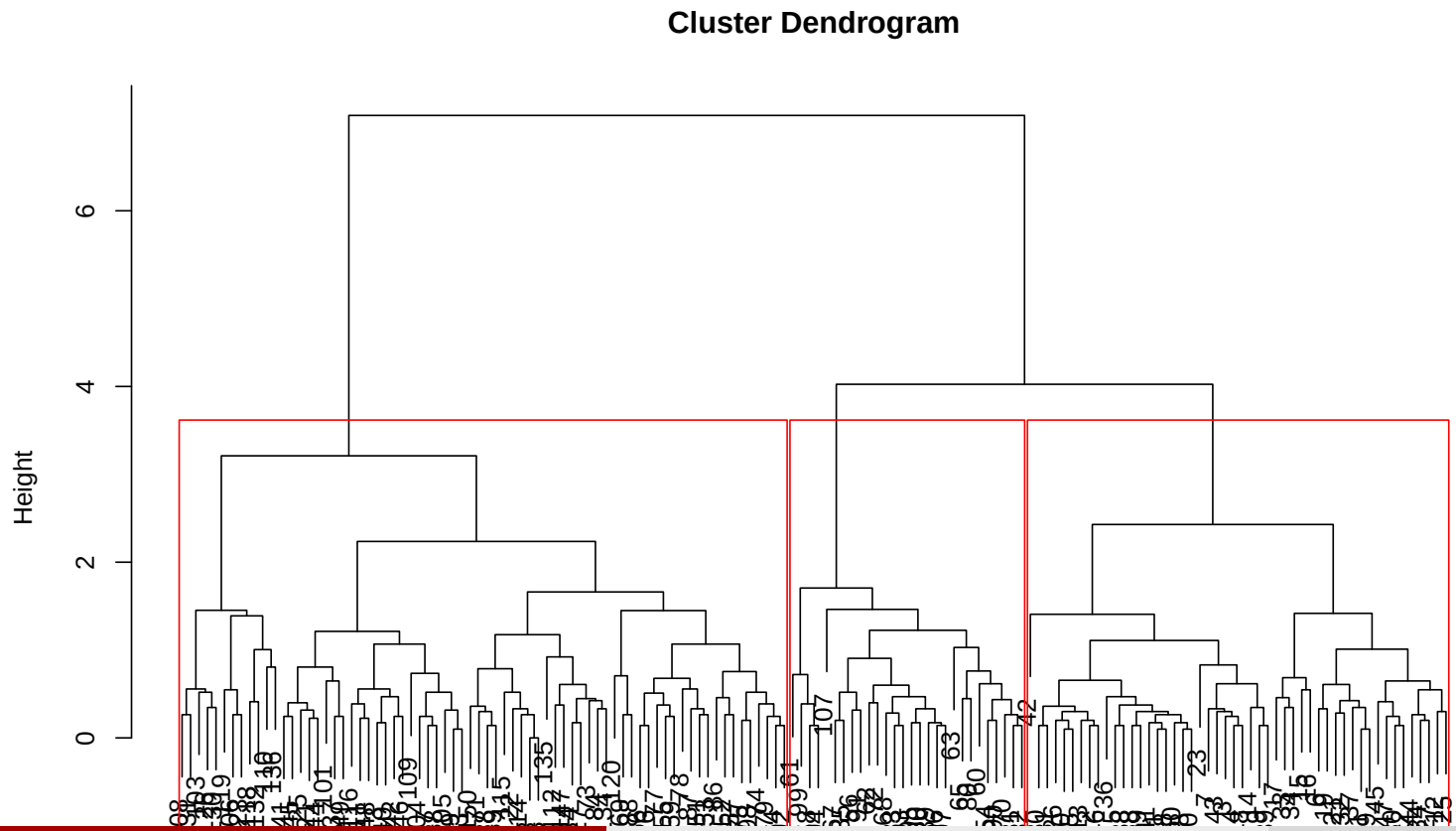


WESTERN SYDNEY UNIVERSITY

Plotting (Continued...)

rect.hclust can be used to box in the clusters.

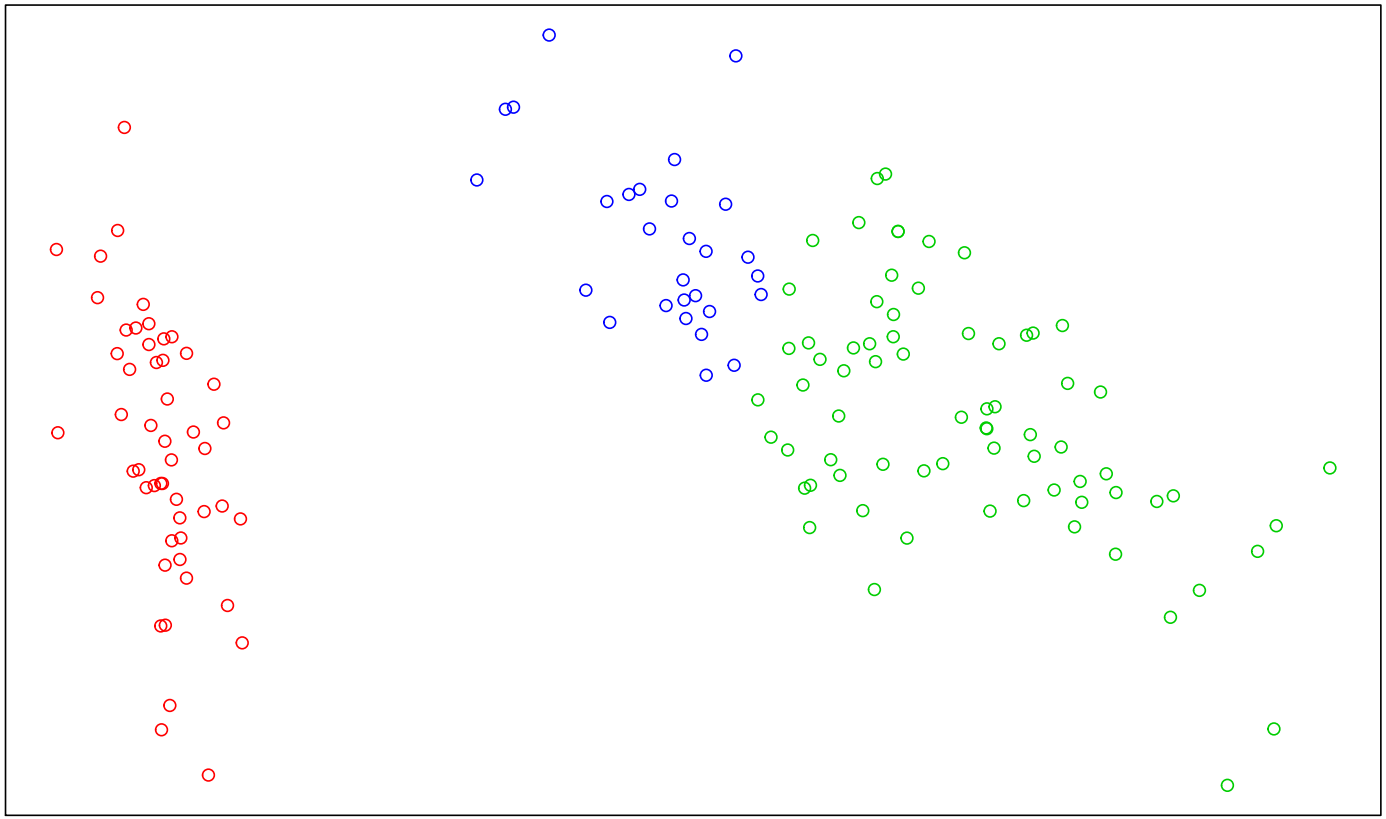
```
plot(hh, xlab=" ", sub = "Complete link cluster analysis")
rect.hclust(hh, k=3)
```



WESTERN SYDNEY UNIVERSITY

Principle Components Again

```
plot(pp$x[,1:2], col=cutree(hh,  k=3)+1,
     xaxt="n", yaxt="n")
```



WESTERN SYDNEY UNIVERSITY

TEXT BOOK

Lecture notes are based on the textbook.

For further reference refer;

Prescribed Textbook - Chapter 10

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R Springer.

WESTERN SYDNEY UNIVERSITY