Data Clustering with R[1]

Yanchang Zhao
http://www.RDataMining.com

R and Data Mining Workshop
for the Master of Business Analytics course, Deakin University, Melbourne

28 May 2015

[1] Presented at Australian Customs (Canberra, Australia) in Oct 2014, at AusDM 2014 (QUT, Brisbane) in Nov 2014 and at UJAT (Mexico) in Sept 2014

Outline

Data Clustering with R [2]

- k-means clustering with kmeans()
- k-medoids clustering with pam() and pamk()
- hierarchical clustering
- density-based clustering with DBSCAN

[2] Chapter 6: Clustering, in book *R and Data Mining: Examples and Case Studies*. `http://www.rdatamining.com/docs/RDataMining.pdf`

Outline

k-means clustering

```
set.seed(8953)
iris2 <- iris
iris2$Species <- NULL
(kmeans.result <- kmeans(iris2, 3))

## K-means clustering with 3 clusters of sizes 38, 50, 62
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     6.850000    3.073684     5.742105    2.071053
## 2     5.006000    3.428000     1.462000    0.246000
## 3     5.901613    2.748387     4.393548    1.433871
##
## 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...
##  [31] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3...
##  [61] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3...
##  [91] 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 1 3 1 1 1 1 1 1 3 3 1 1...
## [121] 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 1 1 1 3...
##
## Within cluster sum of squares by cluster:
## [1] 23.87947 15.15100 39.82097
```

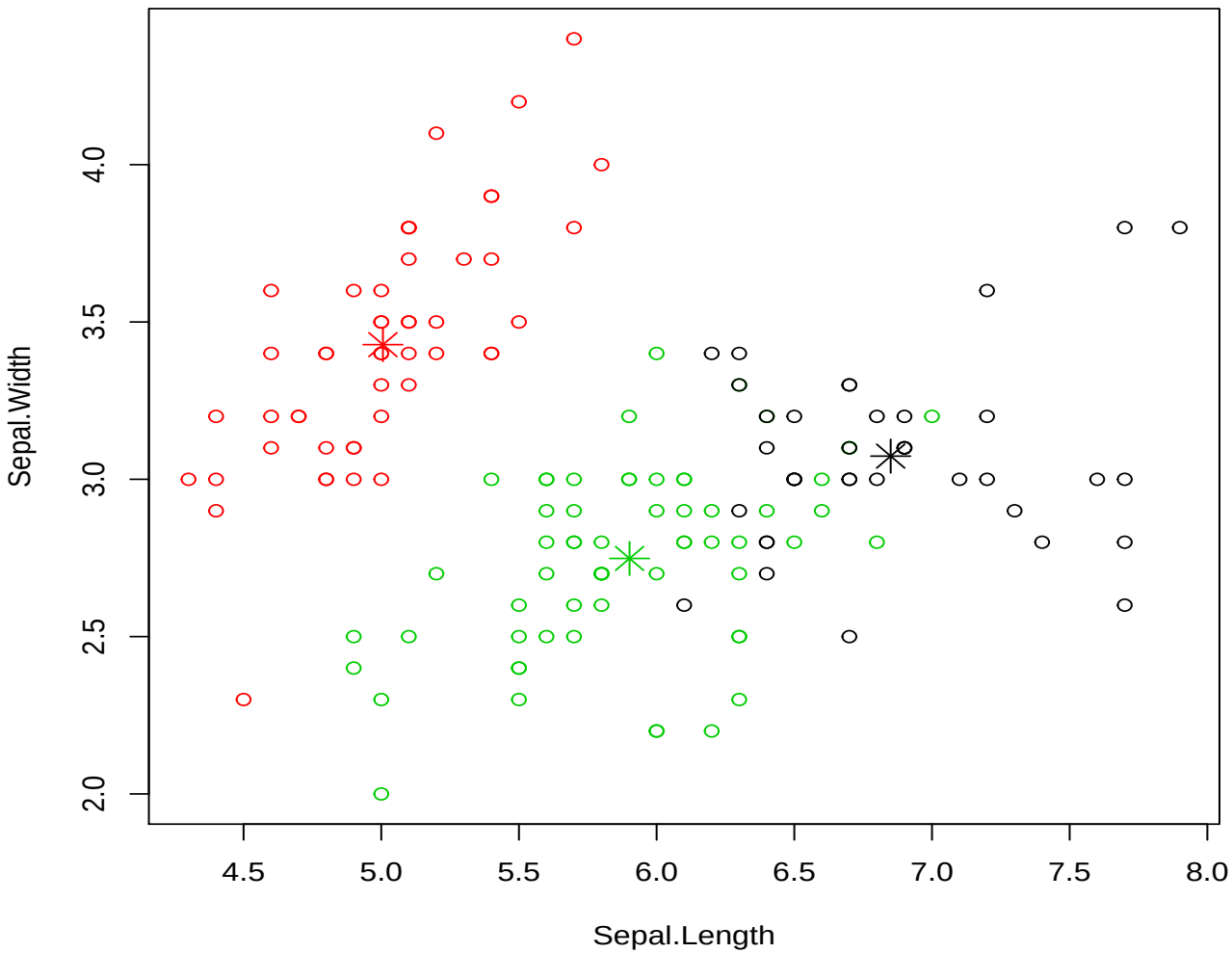
Results of *k*-Means Clustering

Check clustering result against class labels (`Species`)

```
table(iris$Species, kmeans.result$cluster)

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

- Class "setosa" can be easily separated from the other clusters
- Classes "versicolor" and "virginica" are to a small degree overlapped with each other.

```r
plot(iris2[c("Sepal.Length", "Sepal.Width")], col = kmeans.result$clust
points(kmeans.result$centers[, c("Sepal.Length", "Sepal.Width")],
    col = 1:3, pch = 8, cex = 2)  # plot cluster centers
```



Outline

The *k*-Medoids Clustering

- Difference from *k*-means: a cluster is represented with its center in the k-means algorithm, but with the object closest to the center of the cluster in the k-medoids clustering.
- more robust than *k*-means in presence of outliers
- PAM (Partitioning Around Medoids) is a classic algorithm for k-medoids clustering.
- The CLARA algorithm is an enhanced technique of PAM by drawing multiple samples of data, applying PAM on each sample and then returning the best clustering. It performs better than PAM on larger data.
- Functions `pam()` and `clara()` in package *cluster*
- Function `pamk()` in package *fpc* does not require a user to choose *k*.

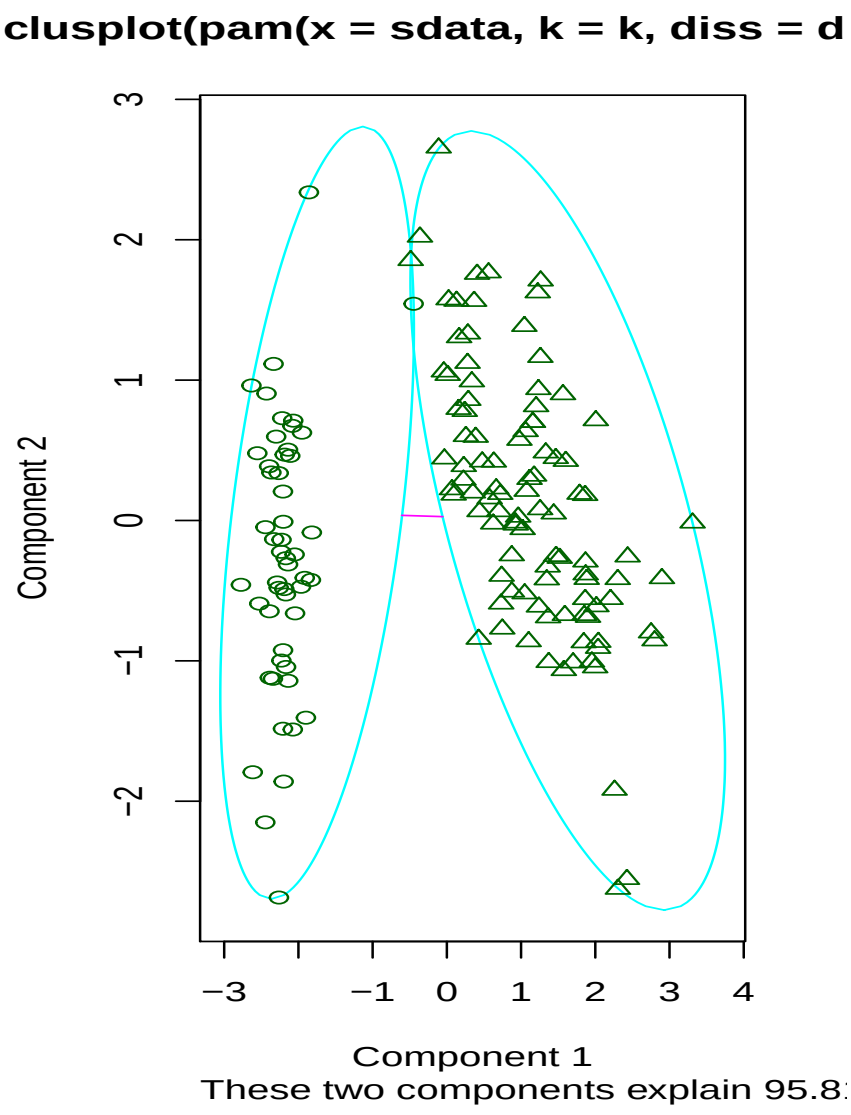
Clustering with pamk()

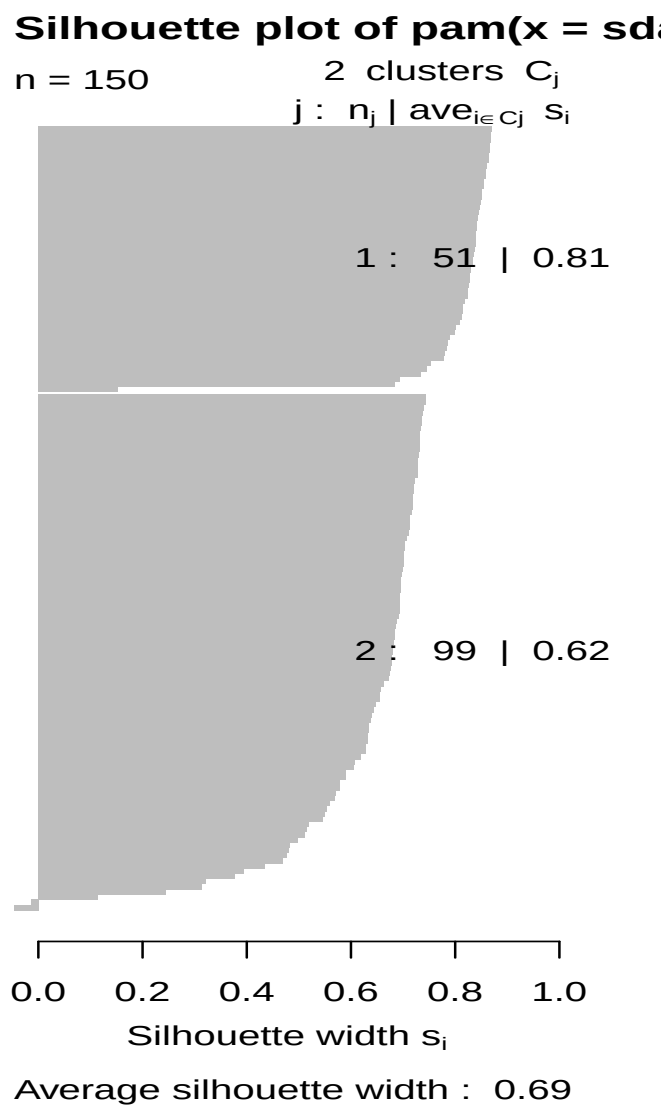
```
library(fpc)
pamk.result <- pamk(iris2)
# number of clusters
pamk.result$nc

## [1] 2

# check clustering against actual species
table(pamk.result$pamobject$clustering, iris$Species)

##
##     setosa versicolor virginica
##   1     50          1         0
##   2      0         49        50
```

Two clusters:

- "setosa"
- a mixture of "versicolor" and "virginica"

```r
layout(matrix(c(1, 2), 1, 2))  # 2 graphs per page
plot(pamk.result$pamobject)
```






```r
layout(matrix(1))  # change back to one graph per page
```

- The left chart is a 2-dimensional "clusplot" (clustering plot) of the two clusters and the lines show the distance between clusters.
- The right chart shows their silhouettes. A large s_i (almost 1) suggests that the corresponding observations are very well clustered, a small s_i (around 0) means that the observation lies between two clusters, and observations with a negative s_i are probably placed in the wrong cluster.
- Since the average S_i are respectively 0.81 and 0.62 in the above silhouette, the identified two clusters are well clustered.

Clustering with pam()

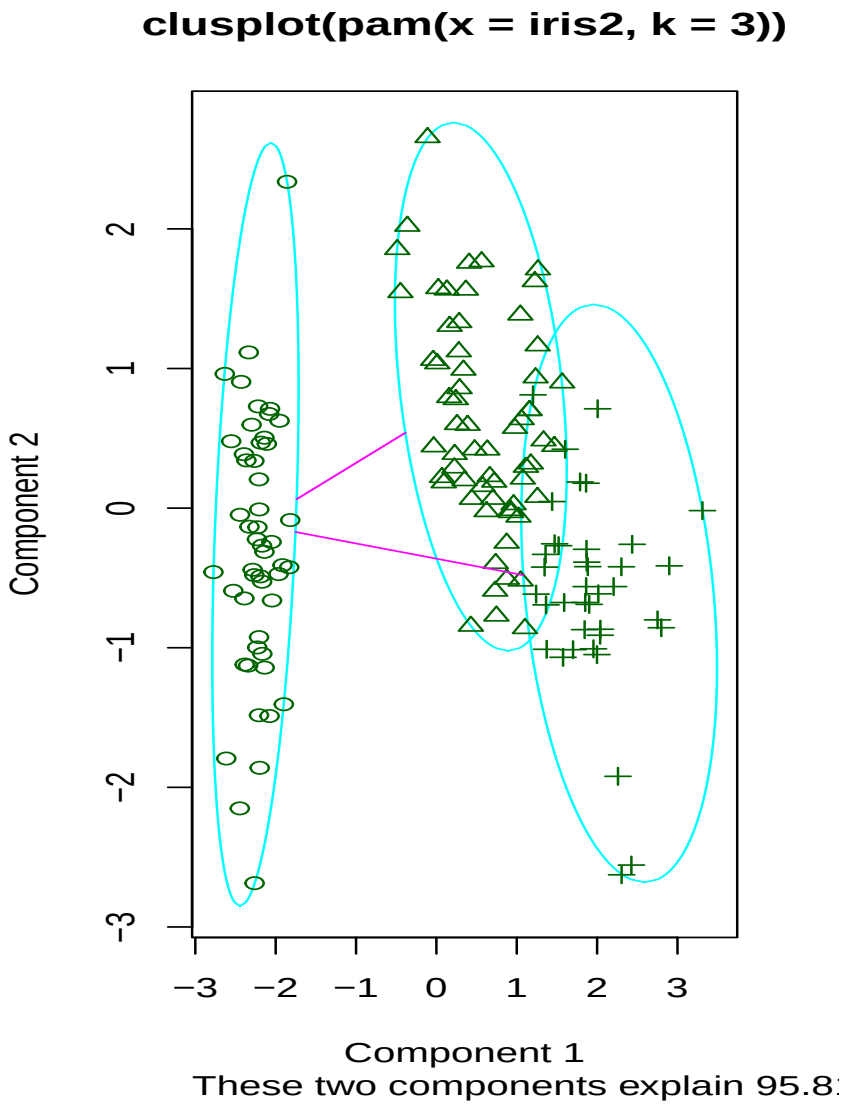
```
library(cluster)
# group into 3 clusters
pam.result <- pam(iris2, 3)
table(pam.result$clustering, iris$Species)
```

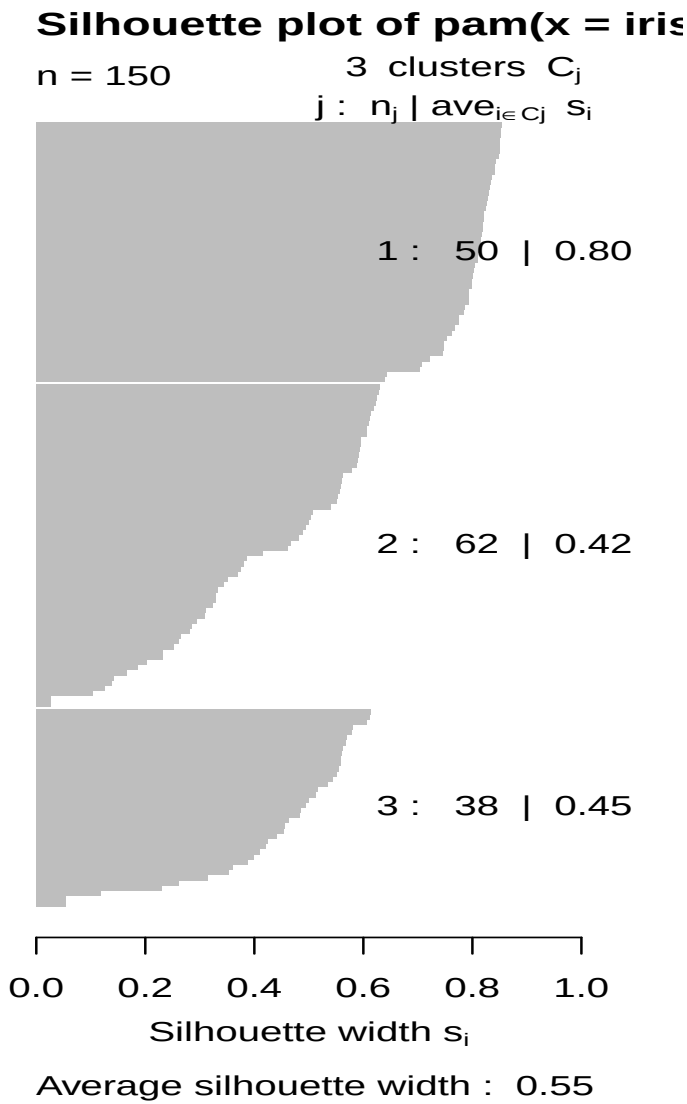
```
##
##      setosa versicolor virginica
##   1     50          0         0
##   2      0         48        14
##   3      0          2        36
```

Three clusters:

- Cluster 1 is species "setosa" and is well separated from the other two.
- Cluster 2 is mainly composed of "versicolor", plus some cases from "virginica".
- The majority of cluster 3 are "virginica", with two cases from "versicolor".

```r
layout(matrix(c(1, 2), 1, 2))  # 2 graphs per page
plot(pam.result)
```






```r
layout(matrix(1))  # change back to one graph per page
```

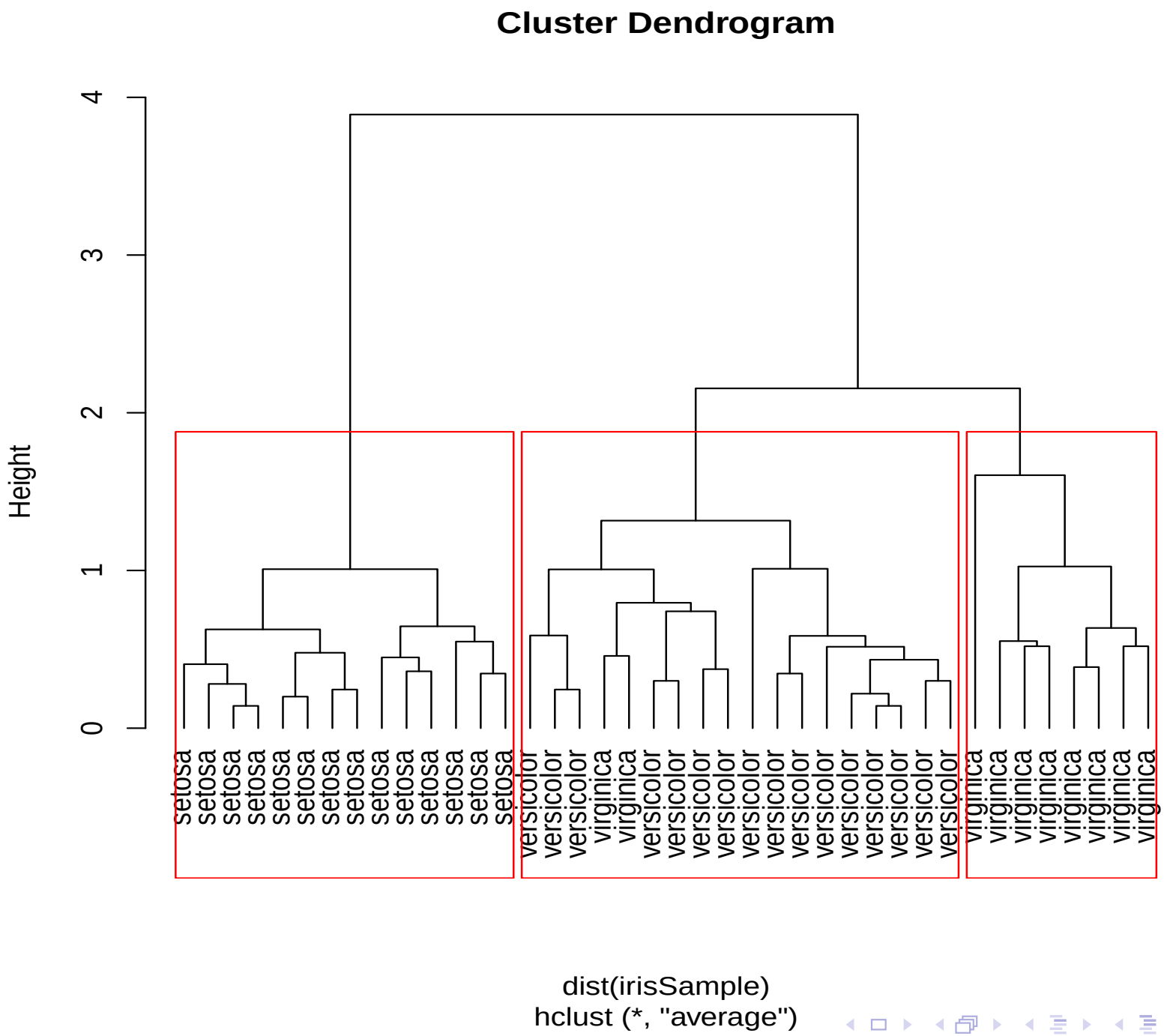
Results of Clustering

- In this example, the result of `pam()` seems better, because it identifies three clusters, corresponding to three species.
- Note that we cheated by setting $k = 3$ when using `pam()`, which is already known to us as the number of species.

Outline

Hierarchical Clustering of the iris Data

```
set.seed(2835)
# draw a sample of 40 records from the iris data, so that the
# clustering plot will not be over crowded
idx <- sample(1:dim(iris)[1], 40)
irisSample <- iris[idx, ]
# remove class label
irisSample$Species <- NULL
# hierarchical clustering
hc <- hclust(dist(irisSample), method = "ave")
# plot clusters
plot(hc, hang = -1, labels = iris$Species[idx])
# cut tree into 3 clusters
rect.hclust(hc, k = 3)
# get cluster IDs
groups <- cutree(hc, k = 3)
```


Cluster Dendrogram
Height
0
1
2
3
4
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
versicolor
versicolor
versicolor
virginica
virginica
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
virginica
virginica
virginica
virginica
virginica
virginica
virginica
virginica
dist(irisSample)
hclust (*, "average")

Outline

Density-based Clustering

- Group objects into one cluster if they are connected to one another by densely populated area
- The DBSCAN algorithm from package *fpc* provides a density-based clustering for numeric data.
- Two key parameters in DBSCAN:
  - `eps`: reachability distance, which defines the size of neighborhood; and
  - `MinPts`: minimum number of points.
- If the number of points in the neighborhood of point α is no less than `MinPts`, then α is a *dense point*. All the points in its neighborhood are *density-reachable* from α and are put into the same cluster as α.
- Can discover clusters with various shapes and sizes
- Insensitive to noise
- The *k*-means algorithm tends to find clusters with sphere shape and with similar sizes.

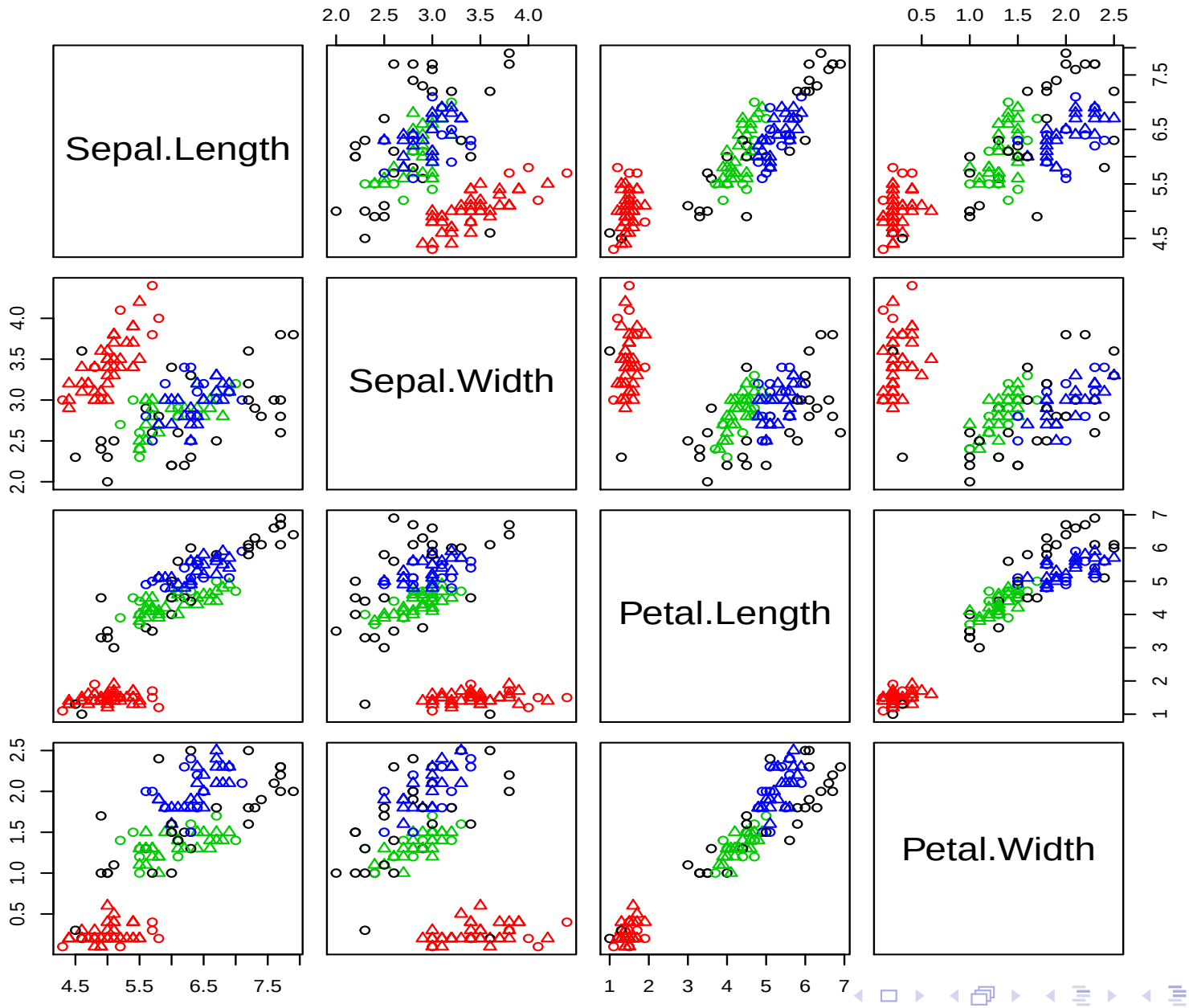
Density-based Clustering of the iris data

```
library(fpc)
iris2 <- iris[-5]  # remove class tags
ds <- dbscan(iris2, eps = 0.42, MinPts = 5)
# compare clusters with original class labels
table(ds$cluster, iris$Species)
```

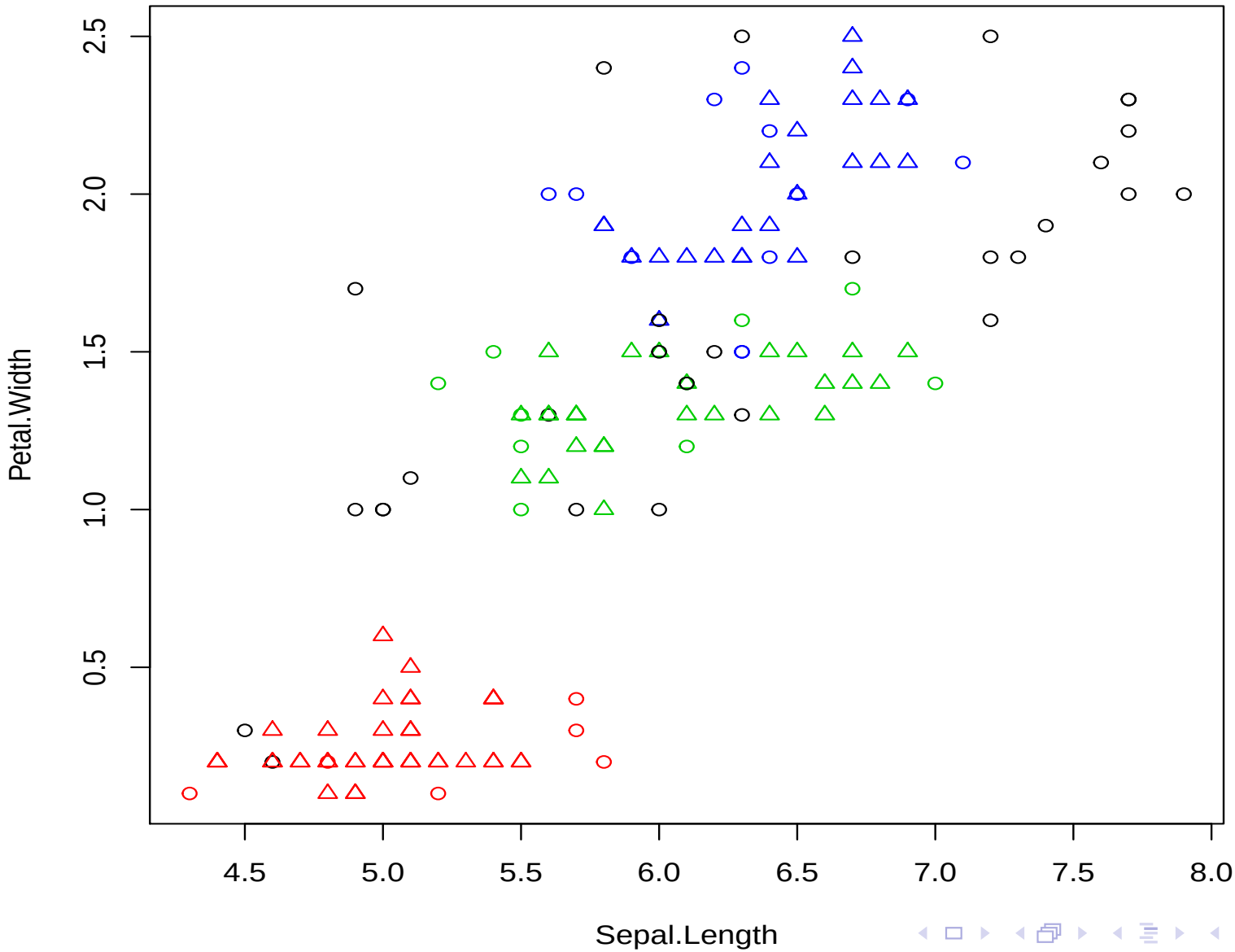
```
##
##     setosa versicolor virginica
##   0      2         10        17
##   1     48          0         0
##   2      0         37         0
##   3      0          3        33
```

- 1 to 3: identified clusters
- 0: noises or outliers, i.e., objects that are not assigned to any clusters

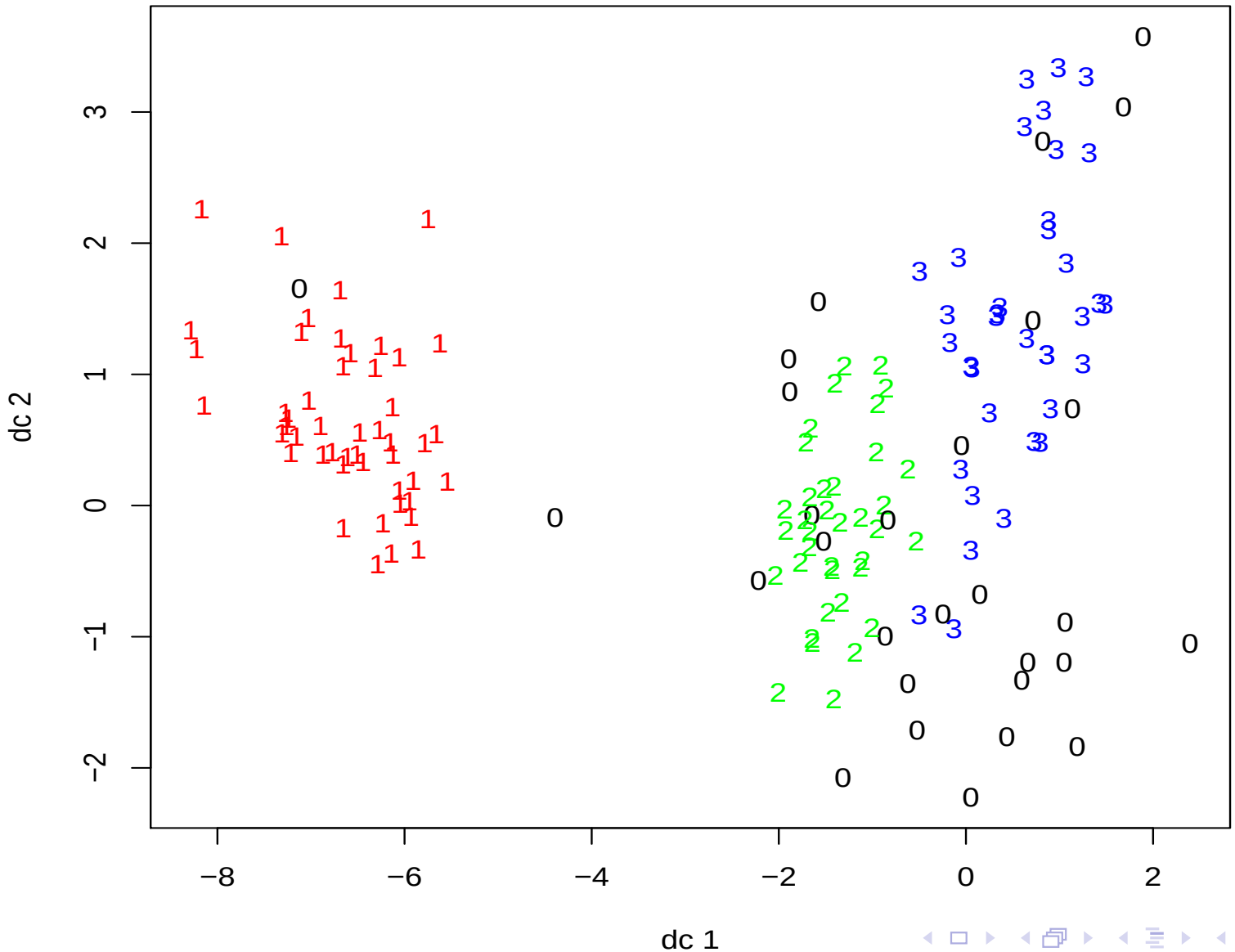
```
plot(ds, iris2)
```



```r
plot(ds, iris2[c(1, 4)])
```



```
plotcluster(iris2, ds$cluster)
```



Prediction with Clustering Model

- Label new data, based on their similarity with the clusters
- Draw a sample of 10 objects from `iris` and add small noises to them to make a new dataset for labeling
- Random noises are generated with a uniform distribution using function `runif()`.

```
# create a new dataset for labeling
set.seed(435)
idx <- sample(1:nrow(iris), 10)
# remove class labels
new.data <- iris[idx,-5]
# add random noise
new.data <- new.data + matrix(runif(10*4, min=0, max=0.2),
                              nrow=10, ncol=4)
# label new data
pred <- predict(ds, iris2, new.data)
```

Results of Prediction

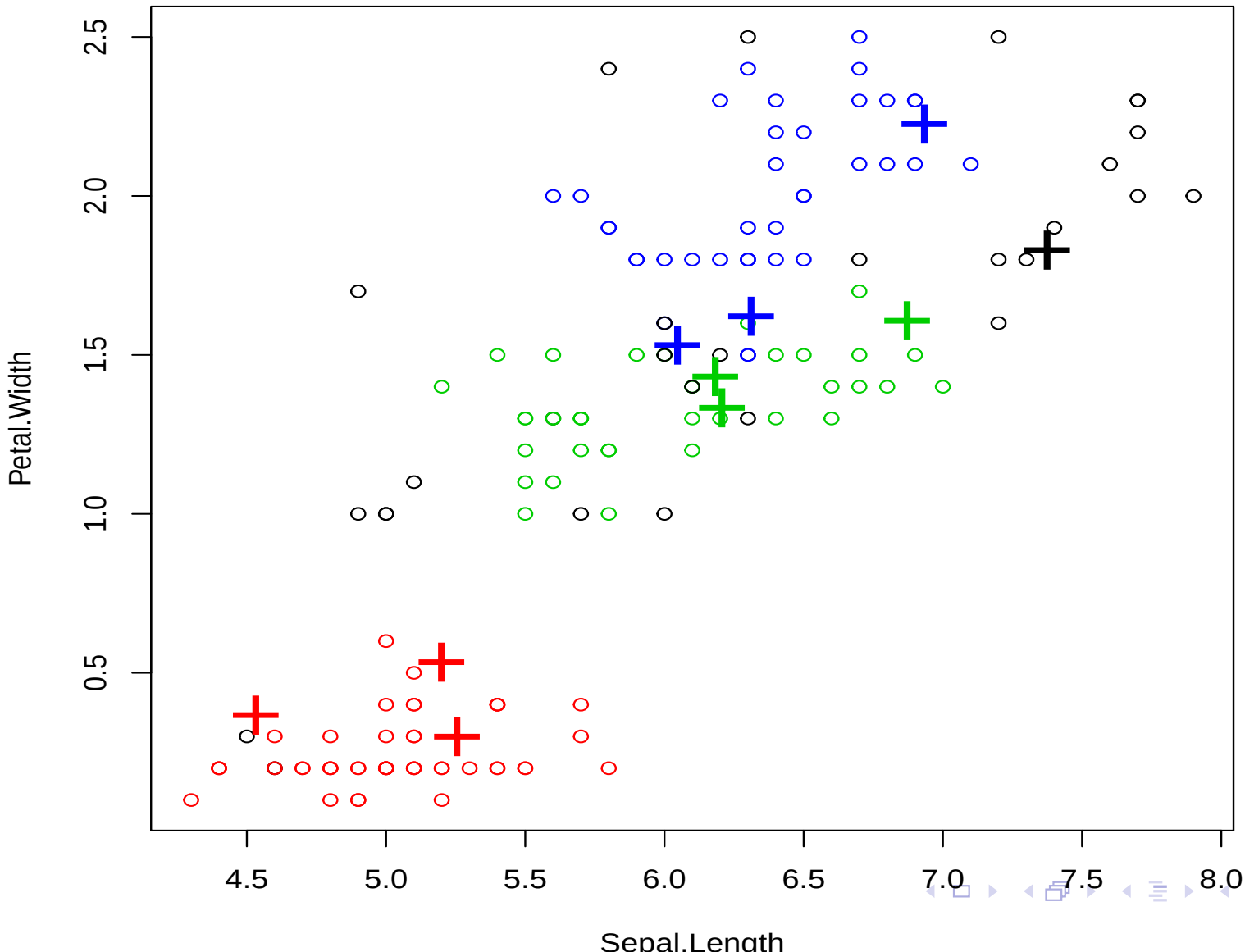
```
table(pred, iris$Species[idx])  # check cluster labels

##
## pred setosa versicolor virginica
##    0      0          0         1
##    1      3          0         0
##    2      0          3         0
##    3      0          1         2
```

Results of Prediction

```
table(pred, iris$Species[idx])  # check cluster labels

##
## pred setosa versicolor virginica
##    0      0          0         1
##    1      3          0         0
##    2      0          3         0
##    3      0          1         2
```

Eight(=3+3+2) out of 10 objects are assigned with correct class labels.

```
plot(iris2[c(1, 4)], col = 1 + ds$cluster)
points(new.data[c(1, 4)], pch = "+", col = 1 + pred, cex = 3)
```



Outline

Online Resources

- Chapter 6: Clustering, in book *R and Data Mining: Examples and Case Studies*

  `http://www.rdatamining.com/docs/RDataMining.pdf`
- R Reference Card for Data Mining

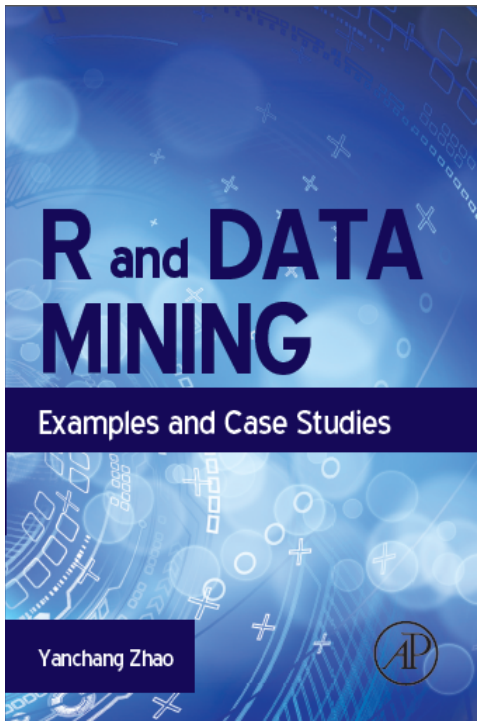
  `http://www.rdatamining.com/docs/R-refcard-data-mining.pdf`
- Free online courses and documents

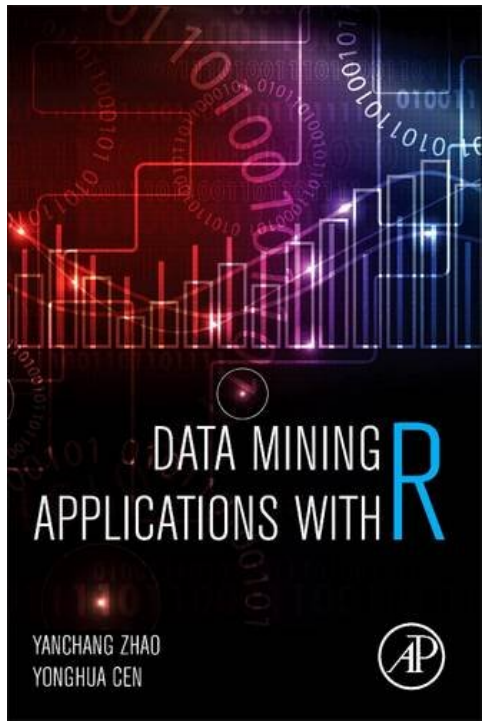
  `http://www.rdatamining.com/resources/`
- RDataMining Group on LinkedIn (12,000+ members)

  `http://group.rdatamining.com`
- RDataMining on Twitter (2,000+ followers)

  `@RDataMining`

The End





Thanks!

Email: yanchang(at)rdatamining.com