Κεφάλαιο 21

Ανάλυση Κατά Συστάδες στην R [1]

21.1 Εισαγωγή

Συστάδα θεωρούμε μια συλλογή από στοιχεία τα οποία είναι όμοια μεταξύ τους (ή βρίσκονται κοντά) και έχουν διαφορές (ή βρίσκονται μακριά) από στοιχεία που ανήκουν σε άλλες συστάδες.

Η ανάλυση κατά συστάδες αποσκοπεί στο διαχωρισμό μιας συλλογής από στοιχεία σε υποσύνολα έτσι ώστε να υπάρχει ομοιογένεια μέσα σε ένα υποσύνολο και ανομοιογένεια μεταξύ των στοιχείων που ανήκουν σε διαφορετικά υποσύνολα. Επιπρόσθετα μπορεί να αποσκοπεί στην ιεραχική οργάνωση των συστάδων με την διαδοχική ομαδοποίηση αυτών, έτσι ώστε σε κάθε στάδιο της ιεραρχίας, οι συστάδες που ανήκουν στην ίδια ομάδα να είναι πιο όμοιες μεταξύ τους από αυτές που ανήκουν σε άλλη ομάδα [1, 2].

Σημαντική έννοια στην ανάλυση κατά συστάδες είναι η απόσταση (ή ομοιότητα), δηλαδή το μέτρο βάση του οποίου δημιουργούνται οι συστάδες. Παραδείγματα μετρικών που μπορούν να χρησιμοποιηθούν ως απόσταση μεταξύ δύο διανυσμάτων $x = (x_1, \ldots, x_p)$ και $y = (y_1, \ldots, y_p)$ είναι:

- Η μετρική Minkowski $d(x, y) = \left[\sum_{i=1}^{p} |x_i - y_i|^m\right]^{\frac{1}{m}}$.
- Για $m = 2$ στην μετρική Minkowski παίρνουμε την Ευκλείδια απόσταση

[1] Το κεφάλαιο στηρίζεται σε ανεξάρτητη εργασία του Α. Ιωάννου

$d(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_p - y_p)^2}$.

- Η μέγιστη απόσταση $d(x,y) = \max\{(x_1 - y_1), \ldots, (x_p - y_p)\}$.
- Η μετρική Canberra $d(x,y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{(x_i - y_i)}$.

Η απόσταση μπορεί να χρησιμοποιηθεί για να κατασκευαστεί πίνακας αποστάσεων σε κάθε στάδιο της ανάλυσης. Ο πίνακας αυτός θα έχει μηδενικά στοιχεία στοιχεία στη διαγώνιο και την απόσταση μεταξύ του i στοιχείου (ή συστάδας) και του j στοιχείου (ή συστάδας) στην θέση (i,j). Ο πίνακας μπορεί να υπολογιστεί στην R με την εντολή `dist(dataset, method)`, όπου `method` κάποια από τις διαθέσιμες μετρικές όπως τις πιο πάνω και `dataset` ο πίνακας των δεδομένων. Όταν δεν προσδιοριστεί κάποια συγκεκριμένη μετρική, χρησιμοποιείται η Ευκλείδια απόσταση.

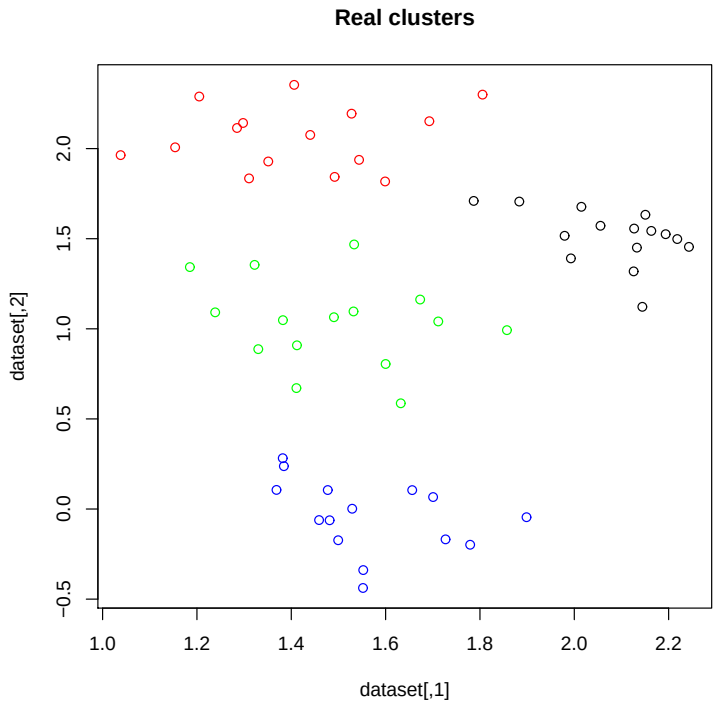
Παρουσιάζουμε παρακάτω ορισμένες μεθόδους ανάλυσης κατά συστάδων. Αρχικά δημιουργούμε ένα σύνολο δεδομένων από 4 γνωστές συστάδες για να ελέγξουμε την αποτελεσματικότητα των μεθόδων ανάλυσης.

```
library(MASS)
library(lattice)
library(cluster)
Sigma<- matrix(c(0.04,0,0,0.04),c(2,2))
x<- mvrnorm(15,c(1.5,2), Sigma)
y<- mvrnorm(15,c(2,1.5), Sigma)
z<- mvrnorm(15,c(1.5,1), Sigma)
w<- mvrnorm(15,c(1.5,0), Sigma)
```

Οι συστάδες θα έχουν 15 στοιχεία η κάθε μια. Τα στοιχεία θα προέρχονται από διδιάστατες κανονικές κατανομές με μέσες τιμές $\mu_1 = (1.5, 2), \mu_2 = (2, 1.5), \mu_3 = (1.5, 1)$ και $\mu_4 = (1.5, 0)$ αντίστοιχα και κοινό πίνακα συνδιακυμάνσεων $\Sigma = \begin{pmatrix} 0.04 & 0 \\ 0 & 0.04 \end{pmatrix}$.

```
x<- cbind(x,1)
y<- cbind(y,2)
z<- cbind(z,3)
w<- cbind(w,4)
data1 <- data.frame(rbind(x,y,z,w))
dataset<- cbind(data1$X1,data1$X2)
distmatrix<- dist(dataset)
mycol <- c("red", "black", "green", "blue")
plot(dataset,col=mycol[data1$X3],main="Real clusters")
```

Συσχετίζουμε κάθε στοιχείο με τον αριθμό της συστάδας από την οποία προέρχεται. Ακολούθως υπολογίζουμε τον πίνακα με τις Ευκλείδιες αποστάσεις για τα δεδομένα για να χρησιμοποιηθεί στη συνέχεια για την ανάλυση σε συστάδες. Τέλος δημιουργούμε το γράφημα των δεδομένων χρησιμοποιώντας διαφορετικό χρώμα για κάθε συστάδα.



Σχήμα 21.1: Πραγματικές συστάδες

21.2 Ιεραρχική Ανάλυση κατά Συστάδες

Προσθετική μέθοδος (Agglomerative Hierarchical Clustering)

Περιγραφή

1. Αρχίζουμε με N συστάδες, με την κάθε μία να περιέχει μόνο ένα στοιχείο και ενα $N \times N$ πίνακα με αποστάσεις.

2. Βρίσκουμε στον πίνακα το ζεύγος U και V συστάδων με την μικρότερη απόσταση μεταξύ τους.

3. Ενώνουμε τις συστάδες U και V σε μια συστάδα, έστω UV. Ανανεώνουμε τον πίνακα αποστάσεων διαγράφοντας τις γραμμές και στήλες που αντιστοιχούν στις U και V και προσθέτοντας μια γραμμή και μια στήλη με τις αποστάσεις της UV από τις υπόλοιπες συστάδες.

4. Επαναλαμβάνουμε τα βήματα 2 και 3 $(N-1)$ φορές μέχρι να υπάρχει μόνο μια συστάδα. Καταγράφουμε τις συστάδες που δημιουργήθηκαν κατά τη διάρκεια της διαδικασίας και το επίπεδο (απόσταση) στο οποίο δημιουργήθηκε η κάθε μία.

Επιλογές για απόσταση μεταξύ συστάδων

(α) Single Linkage: hclust (distmatrix , method="single")
Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την απόσταση με την μικρότερη τιμή από όλες τις πιθανές αποστάσεις μεταξύ ενός στοιχείου (ή συστάδας) του U και ενός στοιχείου (ή συστάδας) του V.

(β) Complete linkage: hclust(distmatrix,method="complete")
Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την απόσταση με την μεγαλύτερη τιμή από όλες τις πιθανές αποστάσεις μεταξύ ενός στοιχείου (ή συστάδας) του U και ενός στοιχείου (ή συστάδας) του V.

(γ) Average linkage: hclust(distmatrix,method="average")
Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την μέση απόσταση μεταξύ των δύο συστάδων (το άθροισμα όλων των πιθανών αποστάσεων μεταξύ ενός στοιχείου του U και ενός στοιχείου του V διά του γινομένου του πλήθους των στοιχείων της U επί του πλήθους των στοιχείων της V).

(δ) Ward's Hierarchical Clustering: hclust(distmatrix,method="ward")
Για κάθε συστάδα k θεωρούμε ως ESS_k το άθροισμα των τετραγώνων των

αποστάσεων κάθε στοιχείου της συστάδας από τον μέσο της συστάδας και ESS το άθροισμα των ESS_k. Ως απόσταση μεταξύ δύο συστάδων U και V θεωρούμε την αύξηση που θα προκύψει στο ESS από την ένωση των δύο συστάδων.

Εφαρμογή

```
hrs<-hclust(distmatrix,method="single")
hrc<-hclust(distmatrix,method="complete")
hra<-hclust(distmatrix,method="average")
hrw<-hclust(distmatrix,method="ward")
```

Εφαρμόζουμε την προσθετική μέθοδο της ιεραρχικής ανάλυσης κατά συστάδες για τις 4 διαφορετικές επιλογές αποστάσεων.

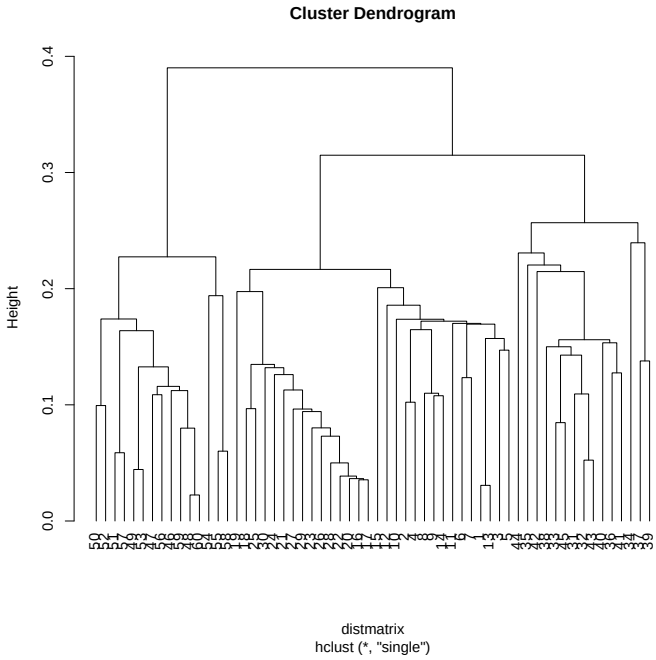
```
membs<- cutree(hrs,k=4)
membc<- cutree(hrc,k=4)
memba<- cutree(hra,k=4)
membw<- cutree(hrw,k=4)
```

Χρησιμοποιούμε την εντολή cutree για να χωρίσουμε τα δεδομένα σε 4 συστάδες. Η εντολή επιστρέφει ένα διάνυσμα μήκους όσο και το πλήθος των δεδομένων,το οποίο έχει τιμές που υποδηλώνουν σε ποια συστάδα ανήκει το αντίστοιχο στοιχείο.

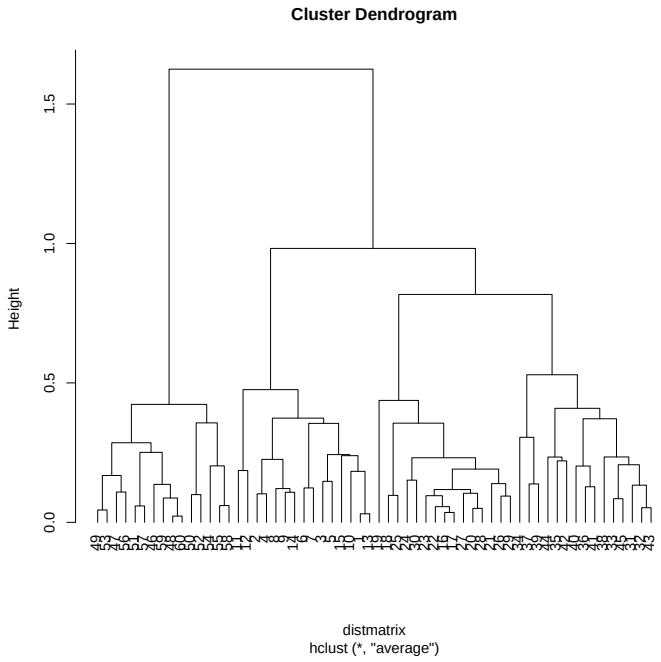
```
> c(sum(membs==1),sum(membs==2),sum(membs==3),sum(membs==4))
[1] 30 12  3 15
> c(sum(membc==1),sum(membc==2),sum(membc==3),sum(membc==4))
[1] 15 15 15 15
> c(sum(memba==1),sum(memba==2),sum(memba==3),sum(memba==4))
[1] 15 15 15 15
> c(sum(membw==1),sum(membw==2),sum(membw==3),sum(membw==4))
[1] 15 15 15 15
> hrs$order
  50 52 51 57 49 53 47 56 46 59 48 60 54 55 58 19 18 25 30 24 21 27 29 23 26
  28 22 20 16 17 15 12 10  2  4  8  9 14 11  6  7  1 13  3  5 44 35 42 38 33
  45 31 32 43 40 36 41 34 37 39
> hrc$order
  49 53 47 56 50 52 54 55 58 51 57 46 59 48 60 11 12  2  4  8  9 14  6  7  3
  5 15 10  1 13 19 30 18 25 21 23 22 26 20 28 27 16 17 24 29 44 35 42 40 32
  43 36 41 33 45 31 38 34 37 39
> hra$order
  49 53 47 56 51 57 46 59 48 60 50 52 54 55 58 11 12  2  4  8  9 14  6  7  3
  5 15 10  1 13 19 18 25 24 30 23 22 16 17 27 20 28 21 26 29 34 37 39 44 35
  42 40 36 41 38 33 45 31 32 43
```

```
> hrw$order
  49 53 47 56 51 57 54 55 58 50 52 46 59 48 60  6  7  3  5  1 13 10 15 11 12
  2  4  8  9 14 18 25 21 26 29 23 22 16 17 27 20 28 19 24 30 34 37 39 44 35
  42 40 36 41 32 43 33 45 31 38
 plot(hrs,hang=-1); plot(hra,hang=-1); plot(hrc,hang=-1); plot(hrw,hang=-1)
```

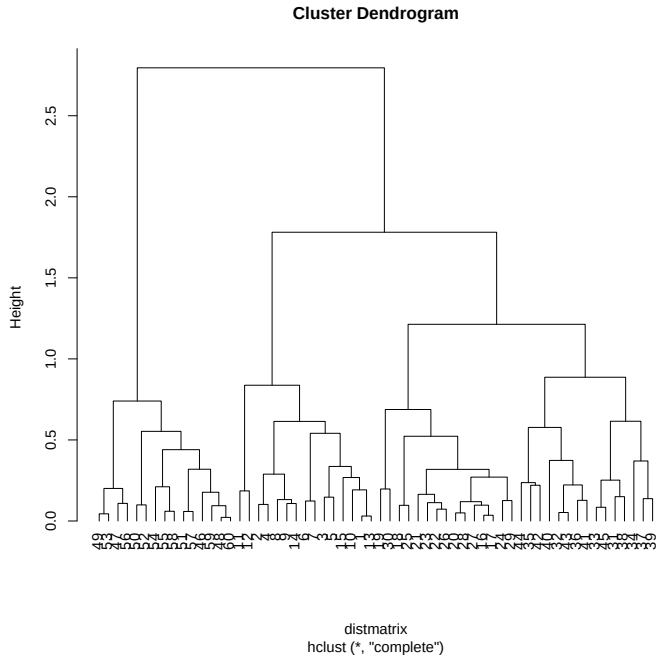
Το αντικείμενο `order` περιέχει σε σειρά την ταξινόμηση των στοιχείων στις συστάδες. Η επιλογή απόστασης `single` δημιουργεί 4 συστάδες οι οποίες δεν αντιπροσωπεύουν τα πραγματικά δεδομένα, ενώ οι υπόλοιπες τεχνικές ομαδοποιούν ορθά τα δεδομένα, όπως φαίνεται και από τα ακόλουθα γραφήματα.



Σχήμα 21.2: Αποτελέσματα ομαδοποίησης από την μέθοδο “Single Linkage”



Σχήμα 21.3: Αποτελέσματα ομαδοποίησης από την μέθοδο “Average Linkage”



Σχήμα 21.4: Αποτελέσματα ομαδοποίησης από την μέθοδο “Complete Linkage”

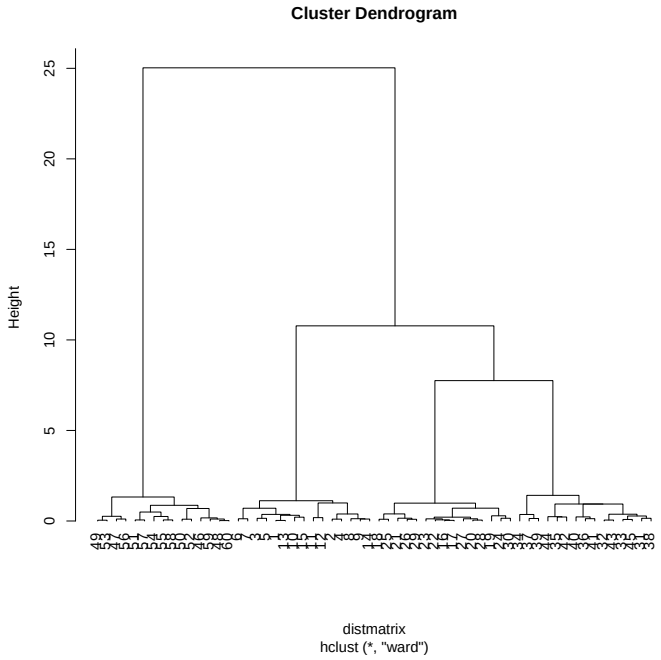
Μέθοδος Διαιρετότητας (Divisive Analysis Clustering-DIANA)

Περιγραφή

diana(distmatrix)

1. Αρχικά επιλέγουμε μια συστάδα.
2. Επιλέγουμε μετά το στοιχείο με τη μεγαλύτερη μέση απόσταση από τα υπόλοιπα στοιχεία της συστάδας, το οποίο γίνεται μια νέα συστάδα.
3. Κατανέμουμε τα στοιχεία της συστάδας είτε στην παλιά συστάδα είτε στην νέα, βάση της απόστασης του κάθε στοιχείου απο τις συστάδες.
4. Επιλέγουμε τη συστάδα με τη μεγαλύτερη διάμετρο (μεγαλύτερη απόσταση μεταξύ δυο στοιχείων της συστάδας) και επιστρέφουμε στο βήμα 2 μέχρι να έχουμε τόσες συστάδες όσα τα στοιχεία μας.

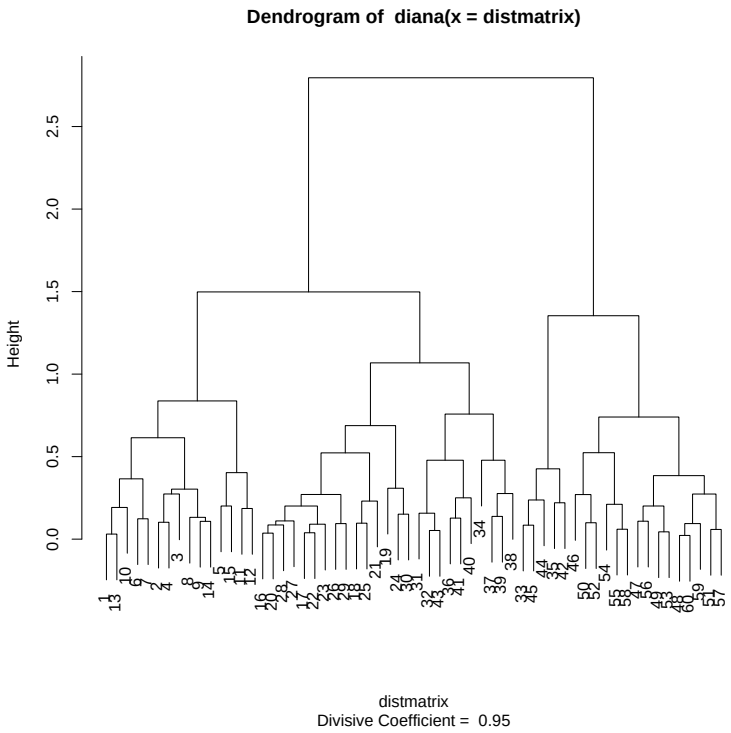
Εφαρμογή



Σχήμα 21.5: Αποτελέσματα ομαδοποίησης από την μέθοδο "Ward's Hierarchical Clustering"

```
> dv<-diana(distmatrix)
> plot(dv)
> dv1 <- cutree(as.hclust(dv), k = 4)
> dv1
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2
[39] 2 2 2 3 2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
> c(sum(dv1==1),sum(dv1==2),sum(dv1==3),sum(dv1==4))
[1] 15 25  5 15
```

Παρατηρούμε ότι η μέθοδος αυτή δεν ομαδοποιεί ορθά τα δεδομένα αφού 10 στοιχεία που θα έπρεπε να είχαν ταξινομηθεί στην τρίτη συστάδα, έχουν τοποθετηθεί στη δεύτερη συστάδα.



Σχήμα 21.6: Αποτελέσματα ομαδοποίησης από την μέθοδο "DIANA"

21.3 Μεθοδολογία K-means (MacQueen)

Περιγραφή

kmeans(dataset, nclusters, algorithm="MacQueen")

1. Επιλέγουμε τυχαία K στοιχεία τα οποία θα αποτελέσουν τους αρχικούς πυρήνες των συστάδων.

2. Για κάθε στοιχείο στα δεδομένα, κατανέμουμε το στοιχείο στην συστάδα της οποίας ο πυρήνας είναι πιο κοντά στο στοιχείο. Οι νέοι πυρήνες (centroids) για τις συστάδες υπολογίζονται ως ο μέσος όρος των στοιχείων της κάθε συστάδας.

3. Επαναλαμβάνουμε το βήμα 2 μέχρι να μην γίνουν αλλαγές στις συστάδες (ή μέχρι ενός ορισμένου αριθμού επαναλήψεων)

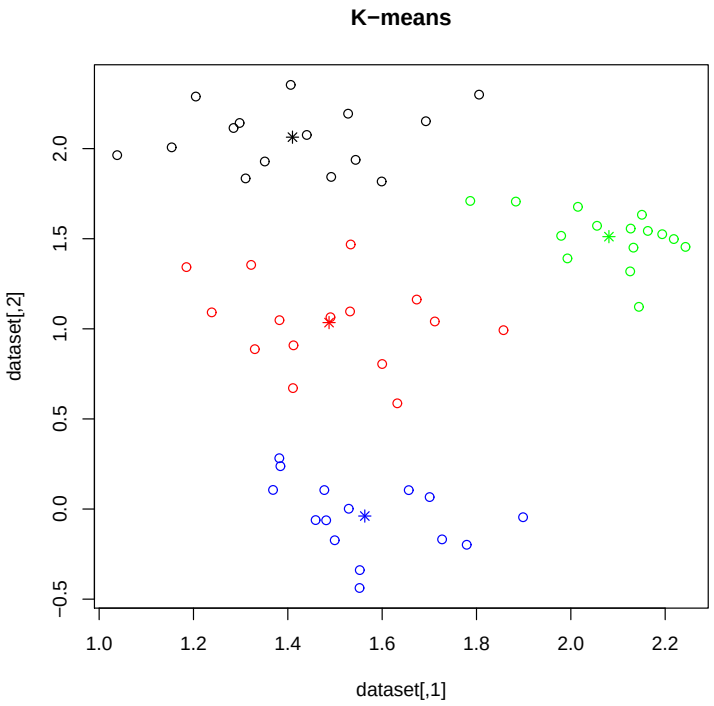
Εφαρμογή

Εφαρμόζουμε την μεθοδολογία K-means επιλέγοντας τον αλγόριθμο `MacQueen`, ο οποίος είναι αυτός που χρησιμοποιείται πιο συχνά για υλοποίηση αυτής της τεχνικής και περιγράφεται πιο πάνω. Χωρίζουμε τα δεδομένα σε 4 συστάδες.

```
>(cl <- kmeans(dataset, 4, algorithm="MacQueen"))
K-means clustering with 4 clusters of sizes 15, 15, 15, 15

Cluster means:
      [,1]        [,2]
1 1.487378  1.03460243
2 1.409862  2.06348506
3 2.080743  1.51151186
4 1.563167 -0.03878101

Clustering vector:
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1
[39] 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

Within cluster sum of squares by cluster:

[1] 1.3218168 1.0320610 0.5656294 0.9120689
```

```
Available components: [1] "cluster"  "centers"  "withinss" "size"

> plot(dataset, col = mycol[cl$cluster], main="K-means")
> points(cl$centers, col = mycol, pch = 8)
```

Παρατηρούμε ότι η μέθοδος ομαδοποιεί ορθά τα δεδομένα σε συστάδες μεγέθους 15 στοιχείων η κάθε μία. Το `Clustering vector` δίνει την συστάδα που ανήκει το κάθε στοιχείο. Τα στοιχεία δημιουργήθηκαν με τέτοιο τρόπο έτσι ώστε τα στοιχεία της κάθε συστάδας να είναι διαδοχικά. Επίσης οι πυρήνες της κάθε συστάδας (`Cluster means`) είναι πολύ κοντά στους πυρήνες που χρησιμοποιήθηκαν για να δημιουργήσουμε στοιχεία για τις 4 συστάδες.



Σχήμα 21.7: Αποτελέσματα ομαδοποίησης από την μέθοδο "K-Means"

21.4 Partitioning Around Medoids (PAM)

Περιγραφή

pam(distmatrix, nclusters)

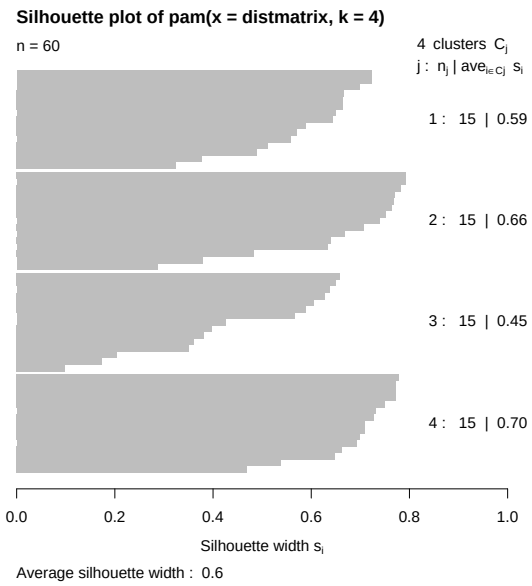
Η μέθοδος Partitioning Around Medoids διαφέρει από την μέθοδο kmeans στο σημείο ότι ως πυρήνας μιας συστάδας είναι πάντα ένα στοιχείο της συστάδας (medoid) και επιδιώκεται η ελαχιστοποίηση της απόστασης των υπόλοιπων στοιχείων από τον πυρήνα. Αρχικά επιλέγεται ένα καλό αρχικό σύνολο από medoids (build phase). Ακολούθως ελέγχεται κατά πόσο η εναλλαγή ενός στοιχείου με ένα medoid θα ελαχιστοποιήσει την απόσταση μεταξύ του πυρήνα και των άλλων στοιχείων και αν ναι, πραγματοποιείται (swap phase).

Εφαρμογή

```
> pamx<-pam(distmatrix,4)
> pamx
Medoids:
     ID
[1,]  3  3
[2,] 22 22
[3,] 32 32
[4,] 59 59
Clustering vector:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
[39] 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
Objective function:
    build      swap
0.2418506 0.2237345


Available components:
[1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
[6] "clusinfo"   "silinfo"    "diss" "call"
> plot(pamx)
```

Παρατηρούμε ότι τα στοιχεία που υποδεικνύει η μέθοδος ως πυρήνες (Medoids) είναι τα στοιχεία 3, 22, 32 και 59 που πραγματικά ανήκουν σε διαφορετικές συστάδες και όλα τα υπόλοιπα στοιχεία ταξινομούνται στην πραγματική τους συστάδα. Επίσης από το Silhouette plot παρατηρούμε ότι δεν υπάρχουν στοιχεία με

αρνητική τιμή, η οποία θα φανέρωνε ότι το αντίστοιχο στοιχείο έχει τοποθετηθεί σε λάθος συστάδα. Τιμές κοντά στο 1 φανερώνουν πολυ καλή ομαδοποίηση ενώ τιμές κοντά στο 0 ότι το στοιχείο βρίσκεται μεταξύ συστάδων.



Σχήμα 21.8: Αποτελέσματα ομαδοποίησης από την μέθοδο "Partitioning Around Medoids"

21.5 Self Organizing Maps (SOM)

Περιγραφή

Η μέθοδος SOM μπορεί να θεωρηθεί ως μια παραλλαγή της μεθόδου kmeans, η οποία περιορίζει τοπολογικά τους πυρήνες των συστάδων. Κάθε μονάδα αντιστοιχεί σε μια συστάδα και ο αριθμός των συστάδων καθορίζεται από το μέγεθος του πλέγματος (ορθογώνιου ή εξαγωνικού σχήματος) πάνω στο οποίο βρίσκονται οι συστάδες.

Αρχικά αναθέτουμε ένα διάνυσμα (codebook vector) σε κάθε μονάδα, το οποίο θα έχει το ρόλο ενός τυπικού μοτίβου συσχετισμένου με τη συγκεκριμένη μονάδα. Συνήθως ένα υποσύνολο των στοιχείων (training set) κατανέμονται τυχαία στις μονάδες. Κατά τη διάρκεια της εκπαίδευσης του αλγορίθμου, τα στοιχεία αυτά παρουσιάζονται επανηλειμμένα, σε τυχαία σειρά, στον τοπολογικό χάρτη. Η μονάδα, η οποία είναι πιο όμοια (winning unit) με το στοιχείο που χρησιμοποιούμε σε κάποιο στάδιο της διαδικασίας, τροποποείται έτσι ώστε η απόσταση της να μειωθεί περαιτέρω από το συγκεκριμένο στοιχείο. Αυτό πραγματοποιείται

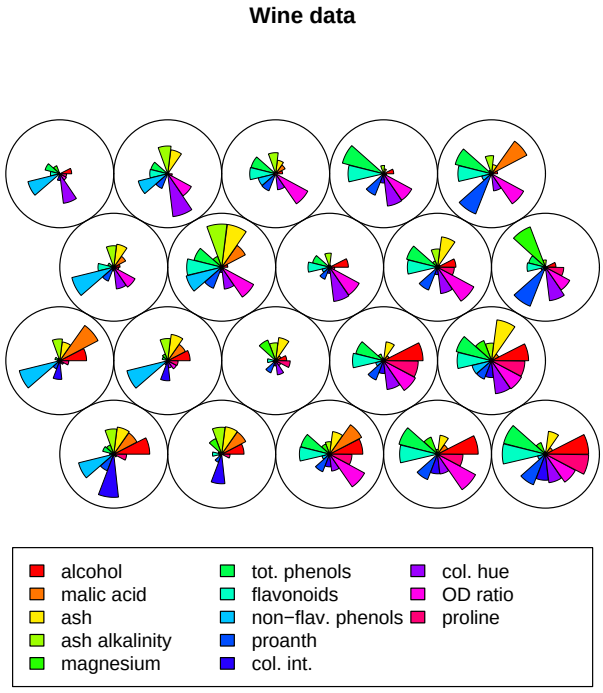
χρησιμοποιώντας σταθμισμένο μέσο όρο, με την βαρύτητα του στοιχείου (learning rate) να είναι μια από τις παραμέτρους της μεθόδου SOM. Συνήθως έχει μικρή τιμή (κοντά στο 0.5). Κατά τη διάρκεια της διαδικασίας, η τιμή αυτή μειώνεται έτσι ώστε ο τοπολογικός χάρτης να συγκλίνει.
Ο περιορισμός τοπολογικά προκύπτει από την απαίτηση του αλγορίθμου γειτονικές μονάδες να έχουν όμοια codebook vectors. Αυτό επιτυγχάνεται τροποποιώντας και τις μονάδες που γειτνυάζουν με την winning unit με τον ίδιο τρόπο. Ο αριθμός των μονάδων που θεωρούνται γειτονικές ως προς την μονάδα αυτή, μειώνεται κατά την εκπαίδευση, έτσι ώστε μετά από ορισμένες επαναλήψεις να τροποποιείται μόνο η συγκεκριμένη μονάδα.

Εφαρμογή

Θα χρησιμοποιήσουμε τα δεδομένα `wines` από το πακέτο `kohonen`, το οποίο παρουσιάζεται στο [3]. Τα δεδομένα αυτά περιέχουν τα αποτελέσματα χημικής ανάλυσης για 177 κρασιά που παράγονται σε μια περιοχή στην Ιταλία και αφορούν 13 χαρακτηριστικά των κρασιών.

```
> library("kohonen")
Loading required package: class
> data("wines")
> wines.sc <- scale(wines)
> set.seed(7)
> wine.som <- som(data = wines.sc,grid=somgrid(5,4,"hexagonal"))
> plot(wine.som, main= "Wine data")
```

Τα αποτελέσματα του σχήματος 9 δείχνουν ότι υψηλά επίπεδα οινοπνεύματος βρίσκονται στα δείγματα κρασιού στην κάτω δεξιά πλευρά του σχήματος, ενώ υψηλή χρωματική συχνότητα βρίσκεται στην κάτω αριστερά πλευρά του γραφήματος.



Σχήμα 21.9: Αποτελέσματα ομαδοποίησης από την μέθοδο “Self Organizing Maps”

21.6 Fuzzy Analysis Clustering (Fanny)

Περιγραφή

fanny(distmatrix, nclusters)

Η μέθοδος ομαδοποίησης Fuzzy επιτρέπει σε κάθε στοιχείο να ανήκει σε περισσότερες απο μια συστάδες. Αυτό επιτυχάνεται υπολογίζοντας κάποια ποσοστά (memberships) για κάποιο στοιχείο για κάθε συστάδα τέτοια ώστε το άθροισμά τους να είναι ίσο με 1.

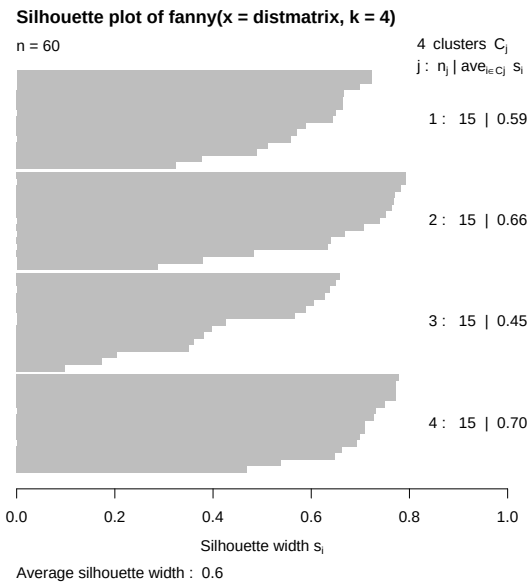
Εφαρμογή

```
> fuzzyc <- fanny(distmatrix,4)
> fuzzyc
Fuzzy Clustering object of class 'fanny' :
m.ship.expon.        2
objective     5.834809
tolerance        1e-15
iterations          13
converged            1
maxit              500
n                   60
Membership coefficients (in %, rounded):
      [,1] [,2] [,3] [,4]
 [1,]   79    9    8    4
 [2,]   66   14   15    6
 [3,]   80    9    7    3
 [4,]   76   10   10    4
 [5,]   71   13   10    5
 [6,]   59   15   18    8
 [7,]   69   12   13    6
 [8,]   54   23   17    6
 [9,]   64   16   14    5
[10,]   66   14   13    7
[11,]   59   21   14    6
[12,]   50   25   16    8
[13,]   79    9    8    4
[14,]   69   15   12    5
```

```
[15,]   66   15   13    6
[16,]    7   83    7    3
.
.
.
[43,]    8   11   74    7
[44,]   11   14   52   23
[45,]   10   11   67   12
[46,]    4    5    9   82
[47,]    6    8   14   72
[48,]    3    4    7   86
[49,]    8   10   21   61
[50,]    8   10   15   66
[51,]    5    7   12   77
[52,]    7    8   13   72
[53,]    9   11   23   57
[54,]    8   11   16   65
[55,]    6    8   13   73
[56,]    5    6   11   79
[57,]    5    7   11   77
[58,]    5    7   11   77
[59,]    3    4    7   87
[60,]    4    4    7   85
Fuzzyness coefficients:
dunn_coeff normalized
 0.5181044  0.3574725
Closest hard clustering:
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3
[39] 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

Available components:
 [1] "membership"  "coeff"       "memb.exp"    "clustering"  "k.crisp"
 [6] "objective"   "convergence" "diss"        "call"        "silinfo"
> plot(fuzzyc)
```

Παρατηρούμε ότι τα `Membership coefficients` είναι αρκετά μεγάλα για την συστάδα στην οποία πραγματικά ανήκουν τα στοιχεία και χαμηλά για τις υπόλοιπες.

Όσο πιο κοντά βρίσκεται ο συντελεστής Dunn(`dunn_coeff`) στο 1 τόσο πιο ξεκάθαρη είναι η ομαδοποίηση των στοιχείων. Από το Silhouette plot παρατηρούμε ότι δεν υπάρχουν στοιχεία με αρνητική τιμή και όλα τα στοιχεία έχουν ταξινομηθεί ορθά.

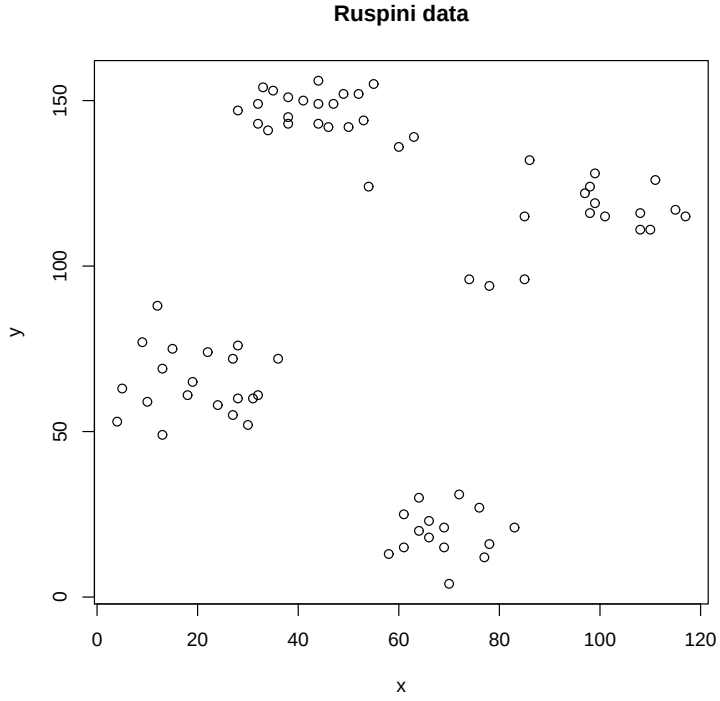


Σχήμα 21.10: Αποτελέσματα ομαδοποίησης από την μέθοδο “Fuzzy Analysis”

21.7 Παράδειγμα ανάλυσης δεδομένων

Θα χρησιμοποιήσουμε το σύνολο δεδομένων `Ruspini` από το πακέτο `cluster`. Τα δεδομένα αυτά είναι χρήσιμα για δοκιμή μεθόδων ανάλυσης κατά συστάδες και περιλαμβάνουν 75 σημεία στον $\mathbb{R}^2$, τα οποία χωρίζονται σε 4 ομάδες.

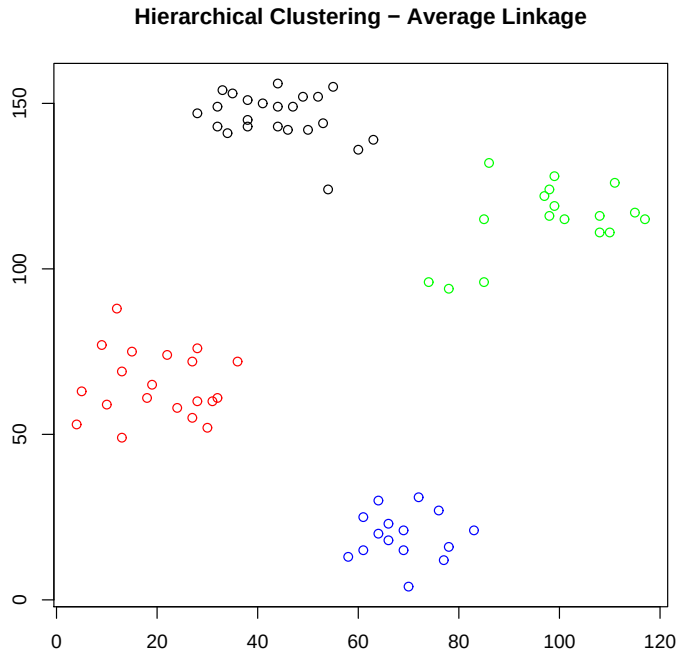
```
> library(cluster)
> data(ruspini)
> plot(ruspini,main="Ruspini data")
> distmatrix <- dist(ruspini)
```



Σχήμα 21.11: Σύνολο δεδομένων Ruspini

```
> mycol <- c("red", "black", "green", "blue")
> hca<-hclust(distmatrix,method="average")
> memba<- cutree(hca,k=4)
> c(sum(memba==1),sum(memba==2),sum(memba==3),sum(memba==4))
[1] 20 23 17 15
> dataset<- cbind(ruspini$x,ruspini$y,memba)
> plot(dataset,col=mycol[memba],main="Hierarchical Clustering - Average Linkage")
```

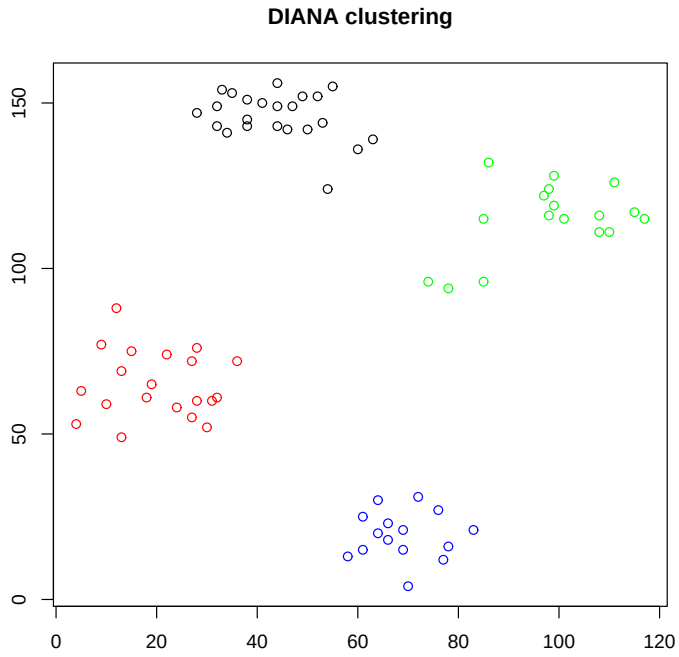
Αρχικά εφαρμόζουμε ιεραρχική ανάλυση κατά συστάδες με average linkage και δημιουργούμε το γράφημα των δεδομένων χρησιμοποιώντας διαφορετικό χρώμα για κάθε συστάδα. Παρατηρούμε ότι η τεχνική εντοπίζει ορθά τις 4 συστάδες.



Σχήμα 21.12: Αποτελέσματα ομαδοποίησης από την μέθοδο “Average Linkage”

```
> dv<-diana(distmatrix)
> dc <- cutree(as.hclust(dv), k = 4)
> dataset<- cbind(ruspini$x,ruspini$y,dc)
> plot(dataset,col=mycol[dc],main="DIANA clustering")
```

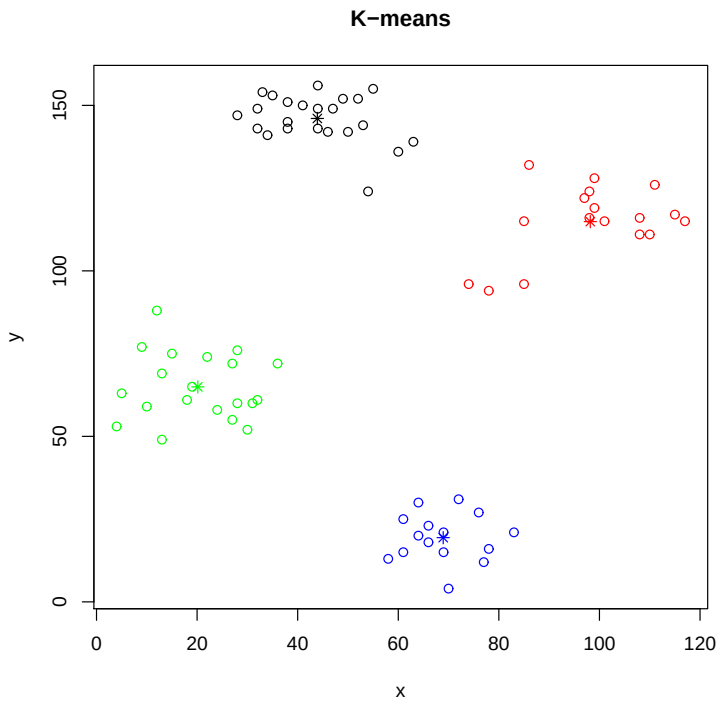
Στη συνέχεια εφαρμόζουμε τη μέθοδο DIANA. Δημιουργούμε το γράφημα των δεδομένων και παρατηρούμε ότι η ομαδοποίηση γίνεται ορθά.



Σχήμα 21.13: Αποτελέσματα ομαδοποίησης από την μέθοδο "DIANA"

```
> cl <- kmeans(ruspini, 4, algorithm="MacQueen")
> plot(ruspini, col = mycol[cl$cluster], main="K-means")
> points(cl$centers, col = mycol, pch = 8)
```

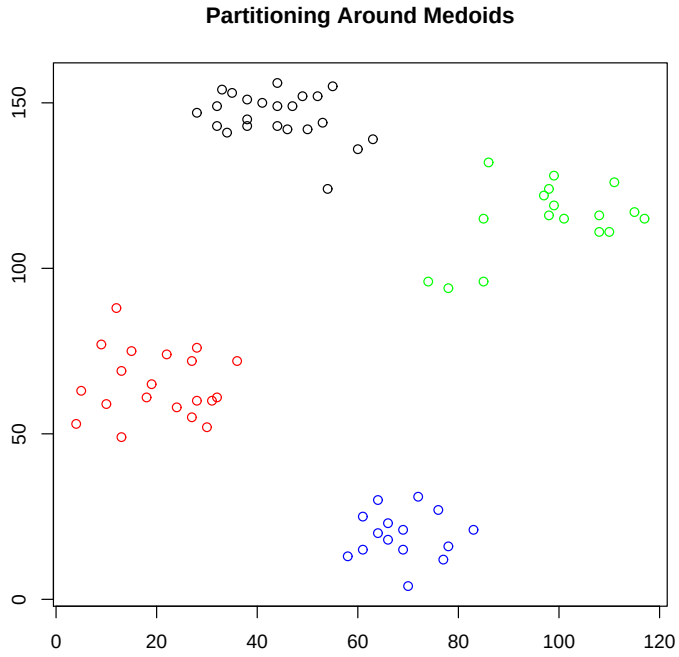
Ακολούθως εφαρμόζουμε τη μέθοδο K-means. Στο γράφημα των δεδομένων βλέπουμε ότι οι συστάδες έχουν επιλεχθεί σωστά, όπως και οι πυρήνες των συστάδων.



Σχήμα 21.14: Αποτελέσματα ομαδοποίησης από την μέθοδο "K-Means"

```
> pamx<-pam(distmatrix,4)
> clusters<-pamx$clustering
> dataset<- cbind(ruspini$x,ruspini$y,clusters)
> plot(dataset,col=mycol[clusters],main="Partitioning Around Medoids")
```

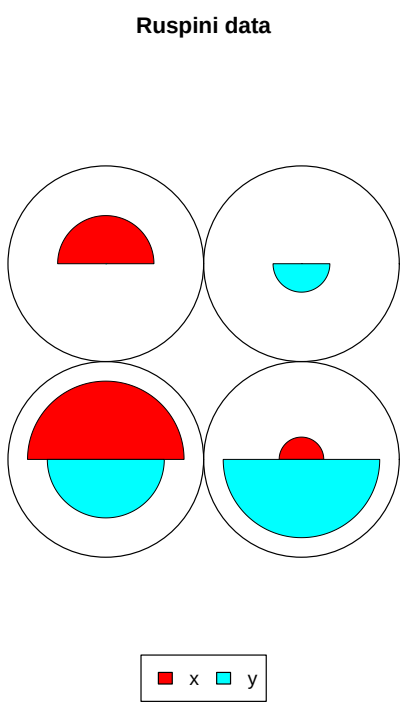
Συνεχίζουμε χρησιμοποιώντας τη μέθοδο Partitioning Around Medoids. Παρατηρούμε ότι και αυτή η τεχνική εντοπίζει ορθά τις συστάδες.



Σχήμα 21.15: Αποτελέσματα ομαδοποίησης από την μέθοδο "Partitioning Around Medoids"

```
> library("kohonen")
> ruspini.sc <- scale(ruspini)
> set.seed(7)
> ruspini.som <- som(data = ruspini.sc,grid=somgrid(2,2,"rectangular"))
> plot(ruspini.som, main= "Ruspini data")
```

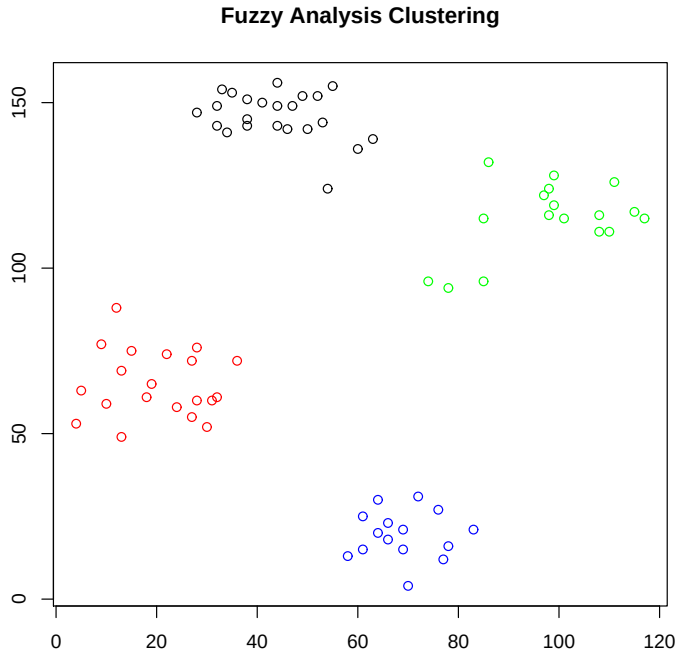
Εφαρμόζουμε επίσης τη μέθοδο "Self Organizing Maps". Τυποποιούμε τα δεδομένα και χρησιμοποιούμε ένα πλέγμα 2×2. Παρατηρούμε ότι η μέθοδος εντοπίζει τα χαρακτηριστικά των 4 συστάδων, δηλαδή ότι η πρώτη συστάδα έχει x κοντά στο 0 και μικρό y, η δεύτερη συστάδα έχει y κοντά στο 0 και μικρό x, η τρίτη συστάδα έχει μεγάλο x και μέτριο y ενώ η τέταρτη συστάδα έχει μικρό x και μεγάλο y.



Σχήμα 21.16: Αποτελέσματα ομαδοποίησης από την μέθοδο "Self Organizing Maps"

```
> fuzzyc <- fanny(distmatrix,4)
> clusters<-fuzzyc$clustering
> dataset<- cbind(ruspini$x,ruspini$y,clusters)
> plot(dataset,col=mycol[clusters],main="Fuzzy Analysis Clustering")
```

Χρησιμοποιούμε τέλος τη μέθοδο ομαδοποίησης “Fuzzy”. Βλέπουμε ότι η επιλογή των συστάδων είναι ορθή.



Σχήμα 21.17: Αποτελέσματα ομαδοποίησης από την μέθοδο “Fuzzy Analysis”

Βιβλιογραφία

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman (2001), The Elements of Statistical Learning, Data Mining,Inference and Prediction, Springer Series in Statistics.

2. Richard A. Johnson, Dean W. Wichern (1998), Applied Multivariate Statistical Analysis, Prentice Hall.

3. Ron Wehrens, Lutgarde M. C. Buydens (October 2007), Self- and Super-organizing Maps in R: The kohonen Package,Journal of Statistical Software, Volume 21, Issue 5.